Protocoles, gestion et transmission sécurisée par chaos des clés secrètes. Applications aux standards: TCP/IP via DVB-S, UMTS, EPS.

Kassem Ahmad

Université Libanaise
École Doctorale
Sciences et Technologies

l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Kassem AHMAD

Mémoire présenté en vue de l'obtention
du grade de Docteur de l'Université Libanaise
du grade de Docteur de l'Université de Nantes
Sous le label de l'Université Nantes Angers Le Mans

Discipline : Electronique
Spécialité : Traitement du signal et des images
Laboratoire : IETR UMR 6164

Soutenue le 16 juillet 2013

École doctorale Sciences et Technologies, Liban

École doctorale Sciences et Technologies de l'Information et Mathématiques (STIM), France
Thèse N° ED503-196

Protocoles, gestion et transmission sécurisée par chaos des clés secrètes. Applications aux standards : TCP/IP via DVB-S, UMTS, EPS.

JURY

Président :	**M. Mohamad KHALIL**, Professeur, Université Libanaise, Liban
Rapporteurs :	**M. Wissam FAWAZ,** Associate Professor, Lebanese Amercican University, Byblos, Liban
	M. Fabrice PEYRARD, Maître de Conférences/HDR, ENSEEIHT/IRIT, Toulouse, France
Examinateurs :	**Mme Ina TARALOVA**, Maître de Conférences, Ecole Centrale de Nantes, France
Directeur de Thèse :	**M. Safwan EL ASSAD**, Maître de Conférences/HDR, Ecole polytechnique de l'université de Nantes, France
	M. Akil JRAD, Professeur, Université Libanaise, Liban
Co-encadrant :	**M. Bassem BAKHACHE**, Maître de Conférences, Université Libanaise, Liban

À mes chers parents, Mohamad et Hassana, sans vous, rien n'aurait pu être possible, que Dieu vous protège et vous prête une longue vie pleine de santé et de prospérité.

À ma sœur et mon frère, Batoul et Hamza, que Dieu nous grade unis pour que nos parents soient toujours fiers de nous.

À tous les membres de ma famille.

Remerciements

Le travail présenté dans cette cotutelle de thèse est le résultat d'une collaboration entre l'Ecole polytechnique de l'Université de Nantes et l'Université Libanaise. Ce travail a été effectué au sein du Laboratoire IETR à Nantes et au sein du laboratoire LASTRE au Liban. Une thèse de doctorat est le fruit d'un travail collectif, pour cela je souhaite adresser mes sincères remerciements aux personnes suivantes :

- Monsieur le Professeur J.F. DIOURIS, Directeur-adjoint de l'IETR, site de Nantes, et Monsieur le Professeur B. EL HASSAN, Directeur du LASTRE au Liban, pour m'avoir accueilli au sein de leur laboratoire et pour m'avoir offert un contexte de travail de très bonne qualité.

- Monsieur le Professeur M. KHALIL, Directeur du Centre Azm pour la recherche en Biotechnologies et ses Applications à l'Université Libanaise, pour m'avoir honoré d'accepter de présider mon jury.

- Monsieur le Professeur F. PEYRARD de l'Institut de Recherche en Informatique de Toulouse IRIT, Toulouse, France ainsi que Monsieur le Professeur W. FAWAZ de l'Université Libano Américaine LAU, Byblos, Liban pour l'intérêt qu'ils ont porté à mon travail de thèse, en acceptant d'être rapporteur.

- Monsieur S. EL ASSAD, Maître de Conférences HDR à l'Ecole polytechnique de l'université de Nantes, et Monsieur A. JRAD, Professeur à l'Université Libanaise, mes Directeurs de thèse pour avoir accepter d'encadrer ma thèse, pour toute leur patience, leur pédagogie et toutes les choses que j'ai appris avec eux. Leur énergie, leur curiosité, et leur dévouement pour la recherche scientifique m'ont beaucoup influencé.

- Monsieur le MCF B. BAKHACHE, mon encadrant de thèse qui m'a accompagné tout au long de ces trois années. Je le remercie pour son aide, ses encouragements, sa grande disponibilité, ses nombreux conseils, et son soutien sans faille qui ont été déterminants pour la réalisation de cette thèse.

- Madame le MCF I. TARALOVA de l'Ecole Centrale de Nantes, Monsieur le Professeur M. KHALIL (encore une fois) de l'Université Libanaise, Tripoli, Liban, pour avoir participé à mon jury de soutenance et pour leurs conseils et corrections.

- Madame S. CHARLIER, Madame L. BUHE et toutes les personnes que j'ai côtoyé au sein du laboratoire IETR, pour leur soutien et aide avant et durant mon séjour au laboratoire.

-Madame Z. IBRAHIM pour son aide dans toutes les tâches administratives à l'Ecole Doctorale EDST de l'Université Libanaise, ainsi que les personnels du centre Azm pour leur appui et leur gentillesse.

- Mes amis qui m'ont aidé beaucoup quand j'ai eu besoin, avec qui j'ai partagé des moments excellents et qui ont fait que mon séjour à Nantes fait partie de mes expériences inoubliables surtout mes amis H. AMMAR et H. BANJAK.

- Mon amour qui m'a apporté le soutien moral, qui m'a encouragé et qui m'a donné l'énergie de continuer dans les moments les plus difficiles.

Sommaire

Introduction générale

Introduction générale 14

1. Contexte de l'étude et généralités

1.1 Concepts de base de la sécurité 21

1.1.1 Sécurité de l'information 21

1.1.2 Principes de conception d'un système de sécurité 22

1.1.3 Entités et canaux de communication 23

1.1.4 Fonctions de sécurité des communications (services et attaques) 23

1.2 Concepts cryptographique de base 25

1.2.1 Fonctions cryptographiques et terminologie 25

1.2.2 Sécurité des systèmes avec des fonctions cryptographiques 26

1.2.3 Chiffrement symétrique 27

1.2.4 Fonctions de hachage avec et sans clé 28

1.2.5 Cryptographie à clé publique et ses composantes 29

1.2.5.1 Certificats numériques 30

1.2.5.2 Algorithmes de chiffrement et de signature asymétriques 30

1.2.5.3 Signatures numériques 32

1.2.6 Cryptanalyse et attaques cryptographiques 33

1.3 Gestion des clés 34

1.4 Chaos 35

1.4.1 Principe du crypto-système basée chaos 35

1.4.2 Propriétés cryptographiques et chaotiques 36

1.4.3 Fonctions chaotiques numériques 36

1.4.4 Solutions pour éviter les effets de la précision finie N 37

1.5 Quelques définitions 38

2. Système de sécurité basé chaos pour les communications IP multicast à travers le DVB-S

2.1 Introduction 41

2.2 Communications IP Multicast à travers le DVB-S 43

2.2.1 Architecture du système DVB-S....43
2.2.2 Structure du RCST émetteur et transmission des paquets IP via DVB-S 44
2.2.2.1 Méthodes d'encapsulation existantes : MPE, IP-Optimized scheme, ULE....45
2.2.2.2 Encapsulation ULE et extension d'entête....46
2.2.2.3 Structure du segment de transport MPEG-2 48
2.2.3 Approches de commutation au niveau satellitaire (Label-Switching et Self Switching) 48
2.2.4 Critères de la sécurité des communications IP par DVB satellitaire....49
2.2.4.1 Attaques actives et passives....50
2.2.4.2 Exigences de la sécurité des données IP via DVB-S....50
2.2.5 Solutions de la sécurité existantes....51
2.2.5.1 Utilisation des entêtes d'extension de l'ULE....51
2.2.5.2 Utilisation d'IPSec en mode tunnel....52
2.2.6 Gestion des clés pour les communications multicast satellitaire 52
2.2.6.1 Systèmes de gestion des clés Flat et LKH 52
2.3 Système de sécurité multicast proposé 54
2.3.1 Présentation générale du système 55
2.3.2 Méthode d'encapsulation proposée EULE....55
2.3.3 Mécanisme proposé pour la sécurité des trames EULE (SEULE)....56
2.3.4 Interfonctionnement entre l'encapsulation SEULE et les approches de commutation .57
2.3.4.1 Encapsulation SEULE avec label-switching 57
2.3.4.2 Encapsulation SEULE avec self-switching 59
2.3.5 Système de gestion de clés proposé TLKH (Two-Tiered LKH) 60
2.3.5.1 Système de dérivation des clés transitoires....61
2.3.5.2 Générateur chaotique proposé pour la génération de clés dynamiques 61
2.3.5.3 Mécanisme d'identification des membres 63
2.3.5.4 Paquet des clés et des paramètres de sécurité proposé (Key PDU)....64
2.3.5.5 Message d'alarme DULM 66
2.3.6 Modification de la structure de l'encapsulateur/décapsulateur ULE 67
2.4 Analyse des performances du système proposé et résultats de simulation 69
2.4.1 Analyse et avantages du système de gestion des clés proposé 69
2.4.1.1 Coût de rekeying (renouvellement des clés)....69
2.4.1.2 Nombre de clés stockées dans les différentes composantes 71

2.4.1.3 Résumé des avantages de TLKH par rapport aux systèmes Flat et LKH 71
2.4.2 Analyse de la consommation de la bande passante 72
2.4.2.1 Données de la gestion des clés 72
2.4.2.2 Taux des données ajoutées par les services de sécurité et de commutation 75
2.5 Conclusions 77

3. Sécurité dans les réseaux mobiles 3G et 4G

3.1 Introduction 81
3.2 Architecture du réseau de troisième génération UMTS 82
3.3 Sécurité dans la troisième génération (UMTS) 84
3.3.1 Principes et objectives de la sécurité de 3G 84
3.3.2 Mécanismes de la sécurité 3G 84
3.3.2.1 Confidentialité de l'identité 84
3.3.2.2 Authentification et établissement des clés AKA (Authentication and Key Agreement) 85
3.3.2.2.1 Vecteur d'authentification AV 86
3.3.2.2.2 Procédure UMTS-AKA 87
3.3.2.2.3 Négociation des algorithmes et validité des clés CK, IK 88
3.3.2.2.4 Procédure d'établissement du mode de sécurité 89
3.3.2.2.5 Protection de l'intégrité et de la confidentialité 92
3.4 Architecture du réseau de quatrième génération EPS 92
3.5 Sécurité dans EPS 94
3.5.1 Principes de la sécurité 4G 94
3.5.2 Exigences de la sécurité en EPS et les menaces principales 95
3.5.2.1 Exigences de la sécurité en EPS 95
3.5.2.2 Menaces contre EPS 96
3.5.3 Architecture de la sécurité en EPS 97
3.5.3.1 Différentes domaines de sécurité 97
3.5.3.2. Sécurité de l'accès au réseau 97
3.5.3.2.1 Confidentialité de l'identité de l'usager et du terminal 97
3.5.3.2.2 Authentification mutuelle entre l'UE et le réseau 98
3.5.3.2.3 Confidentialité des données de l'usager et de la signalisation 98

3.5.3.2.4 Intégrité des données de signalisation 99
3.5.3.3 La sécurité de l'eNB 100
3.5.3.3.1 Démarrage et configuration de l'eNB 100
3.5.3.3.2 Gestion des clés à l'intérieur de la station de base 100
3.5.3.3.3 Traitement des données du plan usager et de contrôle 101
3.5.3.4 Sécurité du domaine réseau 101
3.5.3.5 Sécurité du domaine utilisateur 101
3.5.3.6 Sécurité du domaine application 102
3.5.3.7 Visibilité et configuration de la sécurité par l'utilisateur 102
3.5.3.8 Vue d'ensemble de la sécurité de l'EPS 102
3.5.4 Accès sécurisé au réseau EPS 104
3.5.4.1 Identification des abonnés et des terminaux 104
3.5.4.2 Authentification et établissement des clés EPS-AKA 105
3.5.4.2.1 Génération des vecteurs d'authentification EPS 106
3.5.4.2.2 Procédure EPS-AKA 107
3.5.4.2.3 Distribution des données d'authentification à l'intérieur et entre les réseaux de service 110
3.5.4.3 Hiérarchie des clés 110
3.5.4.3.1 Dérivations et Objectifs des clés dans la hiérarchie 111
3.5.4.3.2 Fonction de dérivation de clés KDF 114
3.5.4.4 Protection de la signalisation NAS, AS et des données usagers 116
3.5.4.4.1 Négociation des algorithmes de sécurité 116
3.5.4.4.2 Protection de la signalisation NAS 117
3.5.4.4.2.1 Etablissement de la sécurité NAS 117
3.5.4.4.2.2 Protection de l'intégrité des messages NAS 118
3.5.4.4.2.3 Chiffrement des messages NAS 120
3.5.4.4.3 Protection de la signalisation AS et des données usagers 121
3.5.4.4.3.1 Etablissement de la sécurité AS 121
3.5.4.4.3.2 Protection de l'intégrité des messages RRC 122
3.5.4.4.3.3 Chiffrement des messages RRC et des données usagers 123
3.6 Conclusion 124

4. Analyse et amélioration de la sécurité de l'EPS

4.1 Introduction 127
4.2. Transmission de l'IMSI en claire 128
4.2.1 Solution d'Al-Saraireh - EMSUCU 129
4.2.2 Solution de Caragata - EEMSUCU 130
4.2.3 Problématique d'EEMSUCU : vulnérabilités et attaques possibles 132
4.2.3.1 Première vulnérabilité et son remède 132
4.2.3.1.1 Remède proposé 133
4.2.3.2 Deuxième vulnérabilité et ses remèdes 133
4.2.3.2.1 Remèdes proposés 134
4.2.3.4 Attaques possibles et proposition d'amélioration 136
4.2.3.4.1 Amélioration proposée 137
4.2.4 Comparaison des coûts d'identification d'un abonné 139
4.3 Analyse des vulnérabilités du protocole EPS-AKA 141
4.3.1 Attaque de déni de service contre l'UE 141
4.3.1.1 Modification des capacités de sécurité d'UE 141
4.3.1.2 Attaque sur le message de rejet du mode de sécurité 142
4.3.1.3 Modification des messages AKA (RAND, AUTN et RES) 142
4.3.2 Attaques contre la clé secrète permanente K 143
4.3.2.1 Attaque sur la voie radio 143
4.3.2.2 Attaque contre la carte à puce UICC 144
4.3.3 Compromis des AV et blocage des services par un Attaque d'homme au milieu (MITM) entre MME et HSS 146
4.3.4 Attaques sur les réponses des données d'authentification (AVs) 147
4.3.4.1 Attaque de l'extérieur 147
4.3.4.2 Attaque de l'intérieur 148
4.3.4.3 Remède contre ces attaques 149
4.4 Protocoles existants et proposés pour remplacer l'EPS-AKA 150
4.4.2 Protocole SE-AKA 150
4.4.2.1 Cryptanalyse du protocole SE-AKA 151
4.4.2.1.1 Attaque par dictionnaire sur l'IMSI chiffré dans SE-AKA 151
4.4.2.1.2 Différentes attaques possibles sur le SE-AKA 154
4.4.3 Protocole EC-AKA 154

4.4.3.1 Cryptanalyse du protocole EC-AKA 155
4.5. Notre protocole proposé FP-AKA 156
4.5.1 Nomenclature 157
4.5.2 Lancement du protocole FP-AKA 158
4.5.3 Composantes de sécurité du protocole FP-AKA 165
4.5.3.1 Génération des nouvelles clés : fonctions et paramètres d'entrée 165
4.5.3.2 Algorithmes et fonctions de sécurité dans FP-AKA 167
4.5.4 Analyse de la robustesse du protocole FP-AKA 167
4.6. Analyse de la qualité de Service des protocoles AKA étudiés 169
4.6.1 Sécurité/Risque 169
4.6.2 Coût 171
4.6.3 Taux de données ajoutées sur la signalisation 172
4.6.4 Délai total de transmission et de traitement 175
4.6.4.1 Délai de transmission 175
4.6.4.2 Délai de traitement au niveau des entités 177
4.6.6 Résumé des résultats 178
4.7 Conclusion 179

Conclusion générale

Annexes

Annexe A : IPsec (Internet Protocol Security) 186
A.1 Association de sécurité SA 186
A.2 Modes : tunnel, transport 187
A.3 Bases des données 187
A.4 IKE 188
Annexe B1 : Les échecs d'authentification 190
Annexe B2 : Différences entre la sécurité UMTS et EPS 191
Annexe B3: L'algorithme à clé publique RSA-OAEP 192
B3.1 Chiffrement par RSA-OAEP 193
B3.2 Déchiffrement RSA-OAEP 195

Références

Liste des sigles

Introduction générale

Introduction générale

Depuis l'antiquité, l'importance de la transmission des messages secrets et sa pratique ont été reconnues. Les méthodes utilisées durant les siècles étaient primitives et leur implémentation était limitée aux besoins de l'armée et de la diplomatie. Pendant la première guerre mondiale, l'absence de chiffrement des transmissions de l'armée russe a permis aux allemands d'intercepter ses messages et de connaître les ordres de déploiement de l'armée russe lors de la bataille de Tannenberg en 1914. Ceci a conduit à la première grande victoire des allemands.

Aujourd'hui, la mondialisation des échanges répandue grâce à l'émergence des nouvelles technologies de l'information et de la communication (Internet, communication par satellite DVB, vidéo conférences, communications sans fil, messagerie électronique,…) pose le problème de la sécurité (confidentialité, authenticité, et intégrité) de l'information transmise à travers les canaux publics non sécurisés. Pour résoudre cette problématique, qui est évolutive, plusieurs travaux de recherche sont déjà réalisés et d'autres sont en cours de réalisation par les chercheurs académiques, privés ou militaires.

L'apport et l'intérêt d'utilisation des signaux chaotiques pour sécuriser les communications numériques sont confirmés par les nombreux travaux internationaux de recherche sur cette thématique. En effet, des caractéristiques importantes des signaux chaotiques, telles que : bonnes propriétés cryptographiques, reproductibilité à l'identique (déterministes), et sensibilité aux conditions initiales et aux paramètres du système, incident à leur utilisation dans les systèmes de communications pour la sécurité des données. La réalisation de nouveaux crypto-systèmes basées chaos est un domaine de recherche relativement récent et il est de plus en plus d'actualité. L'essence des efforts théoriques et pratiques dans ce domaine, découlent du fait que ces crypto-systèmes sont plus rapides que les méthodes classiques, tout en assurant des performances de sécurité, au moins similaires.

Les communications par satellite ont connu un développement continu, parce qu'elles restent la meilleure méthode pour fournir des services de communication dans : des vastes régions géographiques (des continents entiers), des régions/endroits isolé(e)s (les hauts des montagnes, les plates-formes de pétrole, les bateaux) ou dans les régions frappées par une catastrophe (régions affectées par un tremblement de terre ou par une guerre). Aujourd'hui les satellites couvrent une large gamme de services de communication tels que : radio et télévision, fax et téléphone, Internet et transmission des données, etc. Ils ont été envoyés en orbite pour la première fois à la fin des années 1950.

Le standard DVB (Digital Video Broadcasting) qui est très répandu en Asie, en Europe, en Australie et en Afrique a été créé en 1991 pour concevoir des technologies de diffusion de télévision numérique par satellite, par câble et par antenne radio. La famille des standards DVB a été conçue à l'origine pour permettre l'émission de programmes de télévision avec interopérabilité entre les systèmes de communication satellitaires : DVB-S, DVB-S2, DVB-RCS (DVB-Return Channel via Satellite), terrestre : DVB-T (Terrestrial), câblés DVB-C et mobiles : DVB-SH (DVB- Satellite Handheld).

Plus tard, l'évolution des standards DVB, a permis leur utilisation dans les communications IP. Trois méthodes d'encapsulation permettent la transmission des paquets IP par DVB satellitaire,

dont deux méthodes sont des standards à savoir: l'MPE (Multi Protocol Encapsulation) et l'ULE (Unidirectional Lightweight Encapsulation). La troisième méthode « l'IP-Optimized Scheme » est spécifique pour la transmission multicast [Filali *et al.*, 2004], elle est conçu pour permettre l'utilisation des approches de commutation 'label-switching' et 'self-switching'. Parmi ces méthodes, l'ULE a des propriétés supérieures à l'MPE [Collini-Nocker et Fairhurst, 2004] et [Hong *et al.*, 2005] et l'IP-Optimized scheme qui présente des inconvénients et une inefficacité en cas de traitement des deux types de communications, multicast et unicast. Pour cela, dans notre étude, nous ne traitons par la suite que l'encapsulation ULE.

Cependant, deux problèmes restent ouverts concernant l'encapsulation ULE: la sécurité de l'information, et l'adaptation de l'ULE pour le fonctionnement avec les approches de commutation et les technologies OBS (On Board Switching) et spot beam. L'encapsulation ULE, utilise un en-tête réduit au minimum pour avoir un coût faible de traitement, et pour permettre un mécanisme d'extension de l'entête afin d'envoyer des informations requises pour des services additionnels. Néanmoins, il n'y a pas ni une proposition pour le service de commutation satellitaire permettant l'ajout de l'information nécessaire à l'entête ULE pour opérer avec les approches nommées, ni un entête d'extension pour les services de sécurité. Pour pallier au deuxième inconvénient, une proposition a été faite par [Cruickshank *et al.*, 2008], mais la solution préconisée, concerne seulement la structure de l'en tête. Les aspects très importants tels que la gestion des clés secrètes, l'opération avec les technologies et les approches existantes, ont été laissés ouverts.

Un autre problème critique, qui limite l'évolutivité d'un groupe important multicast, est celui de la gestion des clés utilisée dans le système multicast satellitaire. En effet, la gestion des clés intègre un processus de renouvellement de la clé du groupe (Group rekeying) qui s'exécute lorsqu'un membre existant quitte le groupe ou un nouveau membre joint le groupe. Le coût de ce processus est élevé et limite les ressources du réseau lorsque le contrôleur du groupe qui effectue ce renouvellement gère un grand nombre des membres dynamiques du groupe. Par conséquent, ce coût élevé provoque une grande charge sur toutes les composantes du système et une consommation de la bande passante satellitaire qui est coûteuse. Ce problème reste aussi ouvert et sans résolution complète même avec l'utilisation du meilleur système de gestion de clés le plus adapté le LKH (Logical Key Hierarchy) [Howarth *et al.*, 2004].

Dans l'EPS (Evolved Packet System), la 4ème génération des systèmes de communications mobile basées IP, en cours de déploiement autour du globe, les mécanismes de sécurité ont été améliorés par rapport au système de 3ème génération UMTS (Universal Mobile Telecommunications System) afin de résister les nouvelles attaques. Ces nouvelles attaques sont devenues possible à cause : de l'augmentation massive dans la puissance de traitement et la disponibilité d'outils sophistiqués d'une part, et des failles de la sécurité permises par les concepteurs (de 3GPP) de l'architecture de la sécurité EPS d'autre part.

Les spécifications techniques de 3GPP sont mises à jour fréquemment, montrant ainsi un intérêt croissant de la recherche, et de la stratégie pour fortifier EPS. Mais, il existe des vulnérabilités qui n'ont pas encore été résolu dans ces spécifications, même avec les exigences claires indiquées par le groupe de sécurité d'EPS. Ceci est en contradiction avec les exigences spécifiées par ce groupe et la conception de l'architecture de sécurité. Un exemple de ces contradictions est présenté dans [TS 33.401, 2012], où l'exigence affirme que « l'EPS doit être capable de cacher les identités des

utilisateurs auprès des personnes non autorisés », mais dans la conception, l'identité permanente de l'utilisateur se transmet, dans certains cas, en clair sur le canal radio. L'interception de cette identité par l'écoute passive ou par un outil avancé permet le traçage des mouvements des utilisateurs, et ceci peut être utilisé pour provoquer des attaques dangereuses.

La plupart des faiblesses identifiées dans la littérature et celles que nous dévoilons affectent les procédures de sécurisation d'accès qui s'exécutent pour assurer la sécurité des communications. Nous pointons spécialement la procédure d'authentification et d'établissement des clés AKA qui permet/empêcher l'abonnée d'accéder au réseau. Jusqu'au maintenant la plupart des méthodes publiées pour améliorer l'AKA [Xiehua *et al.*, 2011], [He *et al.*, 2008], [Bou Abdo *et al.*, 2012-a] et [Cho *et al.*, 2012] ne sont pas assez robustes. Donc, concevoir un nouvel protocole, apparaît nécessaire pour assurer une protection complète de l'accès au réseau. Ceci est très important et critique pour les communications futures.

Notre travail dans cette thèse consiste à proposer de nouvelles techniques, pour traiter la question de la sécurité des donnés, et la transmission sécurisée des clés secrètes, dans les systèmes: IP via DVB-S et EPS.

Structure de la thèse

Dans le premier chapitre, nous introduisons les concepts de base qui seront nécessaires à la compréhension de la suite de la thèse. Nous commençons par l'introduction des concepts de la sécurité et nous montrons qu'il y a différents domaines pour étudier la sécurité de l'information. Nous présentons les étapes de la conception d'un système de sécurité, les services essentiels de la sécurité que le système doit assurer, et les attaques auxquelles ces services peuvent être soumis. Ensuite, nous introduisons les concepts de base cryptographiques utilisés dans le domaine de la sécurité de l'information et des systèmes, et nous présentons brièvement les principaux types des d'algorithmes cryptographique. Nous présentons brièvement les fonctions de chiffrement symétrique, les fonctions de hachage, les fonctions de chiffrement à clé publique, et nous montrons les différents modèles d'attaques contre le chiffrement et contre la protection de l'intégrité.

Nous présentons aussi la gestion des clés, un sujet important, traité en de détail dans la suite de la thèse. Enfin, nous présentons le principe de crypto-système basé chaos, nous montrons la similarité entre les propriétés chaotiques et la cryptographie, et nous discutons les problèmes liés à la précision finie de la représentation numérique du chaos ainsi que les solutions proposées dans la littérature.

Dans le deuxième chapitre, nous proposons et présentons un système complet pour la sécurité et le transfert efficace des communications IP Multicast à travers le DVB satellitaire. Le système proposé intègre : une méthode d'encapsulation améliorée, efficace du standard ULE, (taux des données ajoutées négligeable), un mécanisme de sécurité qui ajoute un entête d'extension à l'ULE contenant un nonce cryptographique, un système évolutif de gestion des clés à deux couches LKH indépendantes (TLKH), une fonction chaotique de génération des clés secrètes, un algorithme de chiffrement basé chaos, un paquet utilisé pour le transport des clés et des paramètres de sécurité, et un paquet pour la resynchronisation des clés. Nous analysons les avantages du système de gestion des clés TLKH par rapport à d'autres systèmes existants surtout en termes de coût de renouvellement des clés sur la liaison satellitaire. Aussi, nous analysons

notre système du point de vue du taux des données ajoutées par les services de la sécurité et de commutation, et de point de vue de la consommation de la bande passante des données de la gestion des clés par rapport à la meilleure approche.

Dans le troisième chapitre, nous étudions la sécurité dans les réseaux de communications mobiles de troisième génération UMTS et de quatrième génération EPS. Nous commençons par présenter l'architecture du réseau 3G et le fonctionnement de la sécurité dans l'UMTS qui forme la base de la sécurité EPS. Ensuite, nous présentons l'architecture du réseau 4G et la sécurité dans l'EPS. La sécurité dans l'EPS s'inspire de la sécurité dans UMTS, elle utilise les mêmes principes et les éléments robustes de la sécurité 3G (protection de l'identité permanente de l'utilisateur, utilisation d'une carte à puce comme module de sécurité, utilisation des vecteurs d'authentification, authentification mutuelle entre l'utilisateur et le réseau, protection de l'intégrité des données de signalisation, etc.).

Nous présentons les exigences de la sécurité indiquées par le standard d'EPS et les menaces principales contre l'EPS. Nous montrons ensuite l'architecture de la sécurité EPS qui intègre les fonctions de sécurité nécessaires pour répondre aux exigences et pour offrir la sécurité dans tout le réseau. Cette architecture contient cinq domaines d'applications de la sécurité : la sécurité de l'accès au réseau, la visibilité et configuration de la sécurité par l'utilisateur, sécurité du domaine réseau, sécurité du domaine utilisateur, et sécurité du domaine application. La sécurité de l'accès au réseau EPS est la plus importante. Elle contient les fonctions essentielles de sécurité permettant d'apporter des améliorations pour traiter et corriger les faiblesses de la sécurité UMTS. Ces faiblesses sont liées à la confidentialité de l'identité du terminal, l'authentification du réseau de service, la protection de la signalisation NAS (Non Acces Stratum) et la dérivation des nouvelles clés.

L'accès au réseau EPS est le maillon le plus faible de tout réseau sans fil. Pour cela nous concentrons nos efforts sur ce sujet, et nous présentons toutes les composantes nécessaires pour assurer l'accès sécurisé au réseau, à savoir : l'identification des abonnés et des terminaux, la procédure d'authentification et d'établissement des clés (EPS-AKA), la nouvelle hiérarchie des clés, ainsi que l'établissement de la sécurité de la signalisation NAS, AS et des données usagers.

Dans le quatrième chapitre, nous analysons les faiblesses de la sécurité EPS et nous proposons un protocole qui apporte un ensemble d'améliorations permettant de renforcer la sécurité de l'EPS. Nous commençons par l'introduction des menaces non traitées par les fonctions de l'architecture de la sécurité. La transmission en claire de l'identité permanente IMSI est le premier maillon faible (ou menace) de la sécurité EPS que nous traitons. Nous présentons et analysons les solutions proposées par Al-Saraireh et Caragata, et nous ajoutons deux améliorations à la solution de Caragata. Toutes les autres vulnérabilités que nous avons identifiées affectent le protocole standard EPS-AKA. L'exploitation de ces faiblesses conduit à des attaques malveillantes contre les utilisateurs et le réseau. Nous montrons ces différentes attaques (attaque de déni de service contre l'UE, attaques contre la clé secrète K, etc.) et nous présentons les différents protocoles proposés dans la littérature pour les éviter.

Nous analysons ces protocoles : SE-AKA, EC-AKA, et PKBP-SPAKA et nous discutons leurs faiblesses ainsi que les attaques qui peuvent être montées contre eux. Ensuite, nous décrivons le protocole FP-AKA proposé avec toutes ses composantes pour résoudre les faiblesses et prévenir

les attaques identifiées, et nous montrons ses avantages. Enfin, nous effectuons une étude de la qualité de service QoS, afin de comparer les résultats du protocole proposé avec les protocoles de la littérature, en utilisant cinq paramètres de mesure: la sécurité/risque, le coût (CAPEX, OPEX), le taux de données ajoutées, le délai total, la performance.

1. Contexte de l'étude et généralités

1. Contexte de l'étude et généralités

De nos jours, la prolifération des moyens de communications de tous types (réseaux locaux, internet, téléphonies mobiles, etc.) a favorisé l'explosion des communications sur le plan mondial et a introduit la problématique de la sécurité des données échangés. Une grande partie des systèmes de communications utilisent les signaux radio pour porter l'information (téléphonie mobile, communication par satellite, et autres réseaux sans fil). Dans ce contexte, ce n'est pas seulement les utilisateurs autorisés qui peuvent accéder à l'information mais des intrus aussi. Pour empêcher l'accès des intrus, la protection de l'information est une nécessité absolue.

Les services principaux nécessaires pour assurer la sécurité de l'information sont la confidentialité, l'intégrité, et l'authenticité. Ces services sont assurés par une science des techniques mathématiques connue sous le nom de la cryptographie. Cette dernière a été essentiellement utilisée comme moyen de protection des secrets nationaux et stratégiques, et elle a joué un rôle crucial dans plusieurs événements historiques. Nous pouvons dire qu'elle a changé le cours de l'histoire surtout dans les deux guerres mondiales où les praticiens essentiels de cet art étaient les personnels de l'armée et les services gouvernementaux.

Depuis toujours, une compétition se déroule entre celles et ceux qui veulent protéger le secret de leurs communications et celles et ceux qui veulent accéder d'une manière non légale au secret en question. Les techniques et les moyens utilisés actuellement pour assurer la sécurité de l'information ont énormément évolués et des nouvelles solutions de protection sont toujours recherchées.

1.1 Concepts de base de la sécurité

Il n'est pas facile de définir la « sécurité», même si les publics ont tendance à comprendre le sens. Nous pouvons dire qu'elle forme les méthodes de protection contre les actions malveillantes.

De nombreux aspects de la sécurité sont pertinents pour un système de communication. Il y a des aspects de sécurité physique et des aspects de sécurité de l'information. Les premières comprennent les questions des salles verrouillées, des coffres-forts et des gardiens. Dans cette thèse, nous concentrons nos travaux sur les aspects de la sécurité de l'information. En particulier, nous mettons l'accent sur la sécurité des communications.

1.1.1 Sécurité de l'information

Dans le contexte de la sécurité de l'information, les domaines suivants peuvent être étudiés d'une façon indépendante les unes des autres:

- La sécurité du système : Un exemple, est d'essayer de s'assurer que le système ne contient aucune partie faible. Les attaquants essaient d'habitude de trouver un point faible pour le casser.
- La sécurité des applications : Par exemple, les opérations bancaires par Internet utilise en général des mécanismes de sécurité adaptés pour répondre aux exigences spécifiques de l'application.

- La sécurité du protocole : Les parties communicantes sont, par exemple, en mesure d'atteindre les objectifs de sécurité par l'exécution des étapes de communication bien défini dans un ordre aussi bien défini.

- Les primitives de la sécurité : Ce sont les processus de base sur lesquels tous les mécanismes de protection sont construits. Des exemples typiques sont les algorithmes cryptographiques, les fonctions de hachages, mais également des éléments comme une mémoire protégée (carte à puce) peuvent être vue comme une primitive de sécurité.

Dans la conception d'un système de sécurité pratique, il y a toujours des contraintes strictes. Le coût de l'implémentation des mécanismes de protection doit être tempéré avec le montant de risque résolu par ces mécanismes. La convivialité du système ne doit pas souffrir à cause de la sécurité. Ces compromis dépendent aussi de l'utilisation prévue du système. Dans un système militaire, par exemple, les compromis entre la sécurité, le coût et la convivialité se font sur une base différente à celui d'un système de communication à usage général ou public.

1.1.2 Principes de conception d'un système de sécurité

Le processus de conception d'un système de sécurité contient généralement les phases suivantes:

- *Analyse des menaces* : L'objectif est de lister toutes les menaces possibles contre le système, quelle que soit la difficulté et le coût de la réalisation d'une attaque.

- *Analyse des risques* : Le poids de chaque menace est mesuré quantitativement ou, au moins, par rapport à d'autres menaces.

- *Capture des exigences* : Type de protection requis pour le système, tenant compte des phases précédentes.

- *Phase de conception* : Les mécanismes de protection sont conçus, afin de répondre aux exigences de la sécurité, en se basant sur : les blocs de construction existants, tels que les protocoles de sécurité ou primitives ; de nouveaux mécanismes à créer, et une architecture de sécurité à construire. Il est possible que certaines exigences ne puissent pas être satisfaites. Ceci nécessite un retour aux premières phases, en particulier l'analyse des risques.

- *Analyse de la sécurité* : L'évaluation des résultats est habituellement effectuée indépendamment de la phase précédente. Des outils de vérification automatique sont couramment utilisés permettant seulement une analyse partielle de la sécurité. Il y a souvent des faiblesses dans le système de sécurité qui ne peuvent être révélées que par la pratique et l'utilisation des méthodes créatives.

- *Phase de réaction* : Dans cette phase, il est primordial que la conception originale du système soit assez flexible et permette les améliorations. Il est utile d'avoir une certaine marge de sécurité dans les mécanismes. Ces marges ont tendance à être utiles dans le cas où de nouvelles méthodologies d'attaques plus efficace apparaissent.

1.1.3 Entités et canaux de communication

Un système de communication sur lequel on doit appliquer la sécurité est composé au minimum de deux entités et d'un canal de transmission. Les notions utilisées dans ce système sont :

- Une *entité* : est une personne, organisation, ou dispositif qui envoie, reçoit ou manipule l'information.
- Un *émetteur* : est une entité légitime qui envoie l'information.
- Un *destinataire* (ou *récepteur*) : est une entité légitime qui reçoit l'information.
- Un *attaquant* (ou *adversaire*) : est une entité qui essaie de contrecarrer les mesures de sécurité. Les actions de l'attaquant peuvent être très variées en fonction de ses intentions et du système de communication utilisé. L'attaquant essaye par exemple de se faire passer comme le destinataire ou comme l'émetteur d'un message.
- Un *canal* de communication : est un media de transmission de l'information d'une entité à une autre.
- Un *canal physiquement sécurisé* : est un canal qui n'est pas physiquement accessible à l'attaquant.
- Un *canal public sécurisé* : est un canal qui n'est pas normalement accessible à l'attaquant par des moyens cryptographiques.

La figure 1.1, montre l'exemple d'un système de communication entre deux entités *A* et *B*, en présence d'un adversaire qui scrute le canal de transmission pour intercepter les messages transmis.

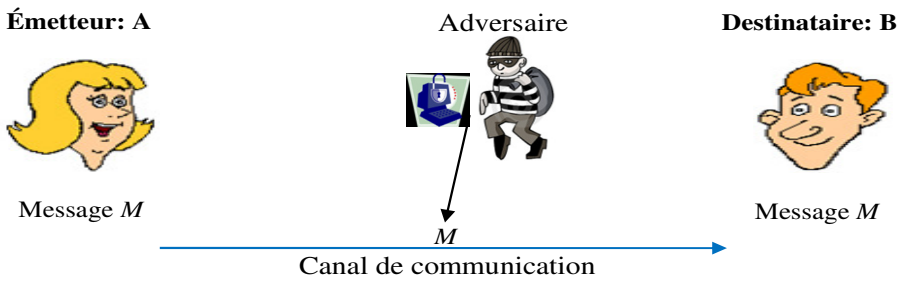

Figure 1.1. Système de communication composé de deux entités et un adversaire

1.1.4 Fonctions de sécurité des communications (services et attaques)

Dans ce paragraphe, nous énumérons les services les plus importants pour assurer la sécurité des communications:

- Authenticité : Dans un scénario classique où les parties *A* et *B* se communiquent sur un certain canal, les deux doivent commencer par identifier les uns les autres. L'authentification est le processus de vérification des identités.

- Confidentialité : Dans le même scénario classique, les partis *A* et *B* peuvent vouloir limiter l'intelligibilité de la communication juste pour les deux parties elles-mêmes (et pas pour quiconque : un attaquant par exemple), pour garder la communication confidentielle.

- Intégrité : Si les messages envoyés par le parti *A* sont identiques à ceux reçus par le parti *B* et vice versa, alors l'intégrité des communications est préservée. C'est la propriété que le message n'a pas été modifié au cours de sa transmission.

- Non-répudiation : Il est souvent utile pour le destinataire *B* de stocker un message reçu de l'émetteur *A*. La non-répudiation du message veut dire que *A* ne peut pas nier plus tard avoir envoyé ce message.

- Disponibilité: Les services et le canal de communication doivent être disponibles à tout moment pour les partis *A* et *B*.

Les attaques typiques contre les services de la sécurité sont :

- Authentification : un imposteur essaye de se faire passer comme l'un des partis communicants.

- Confidentialité : un intrus essaye d'écouteur le canal pour obtenir le contenu de l'information utile. Pour empêcher cela, le service ou la fonction de confidentialité doit être assurée par le chiffrement de l'information comme indiqué dans la figure 1.2.

- Intégrité : un troisième parti (homme au milieu) essaye de modifier, insérer ou supprimer des messages sur le canal de communication.

- Non-répudiation : l'émetteur d'un certain message peut parfois tirer bénéfice s'il peut nier plus tard son émission. C'est le cas par exemple, de message se rapportant à une transaction financière ou un engagement d'achat ou de vente de quelque chose.

- Disponibilité : une attaque de déni de service (DoS) essaye d'empêcher l'accès au canal de communication, au moins pour l'un des partis communicants.

Dans cette thèse, nous allons mettre l'accent sur les quatre services suivants : l'authenticité, la confidentialité, l'intégrité, et la disponibilité. La fonction de non-répudiation est moins critique dans les réseaux satellitaires et EPS. Elle est par contre pertinente pour la couche application.

1.2 Concepts cryptographique de base

La cryptologie est définie comme l'art et la science de l'écriture secrète. La possibilité d'appliquer la cryptologie pour assurer la sécurité des communications est évidente. La cryptologie regroupe :

- la cryptographie: technique de communication secrète de l'information, science relative à la sécurité des messages;
- la cryptanalyse : analyse des systèmes cryptographiques afin de trouver des faiblesses.

Nous allons voir dans la suite les fonctions cryptographiques, qui sont utilisées pour assurer la sécurité d'un système, et les modèles de cryptanalyse possibles par les attaquants. La cryptanalyse essaie de casser les systèmes existants, et recherche constamment sur de nouveaux moyens pour attaquer les systèmes. La cryptanalyse contribue indirectement à la conception de nouveaux systèmes encore plus performants.

1.2.1 Fonctions cryptographiques et terminologie

Nous présentons ci-dessous quelques définitions liées à la cryptographique.

- *Espace de texte en clair ou message M,* est un sous-ensemble de l'ensemble des chaînes binaires $\{0,1\}^*$. Par exemple chaque lettre de l'alphabet français peut être assignée à un mot de cinq bits (codage binaire).

- *Espace de texte chiffré (ou cryptogramme) C,* est également un sous-ensemble de $\{0,1\}^*$.

- *Espace des clés K* est aussi un sous-ensemble de $\{0,1\}^*$. Souvent $K = \{0,1\}^k$ où k est un paramètre de sécurité fixe représentant la clé.

- Fonction (ou algorithme) de *chiffrement E*, permet de calculer C avec différentes fonctions E, $C=E_k(M)$.

- Fonction (ou algorithme) de *déchiffrement D,* permet de retrouver le message en clair à partir du cryptogramme $M = D_k(C)$.

- Un *crypto-système* est un procédé pour transformer un texte clair en un texte chiffré et inversement, il se compose de tous ce qui précède, c.à.d. (M, C, K, E, D).

La figure 1.2, montre le schéma de principe d'un crypto-système qui assure la confidentialité des communications entre Alice et Bob. Les deux entités commencent d'abord par l'échange d'une clé secrète par un moyen donnée. Si Alice veut envoyer un message M à Bob, elle chiffre M avec la fonction E et la clé secrète k et l'envoie à Bob. Après avoir reçu le message chiffré C, Bob effectue le déchiffrement de ce message par la clé secrète déjà partagée et la fonction D afin de recouvrir le message original M. L'attaquant peut intercepter le message C, et peut utiliser des techniques de cryptanalyse pour trouver le message M.

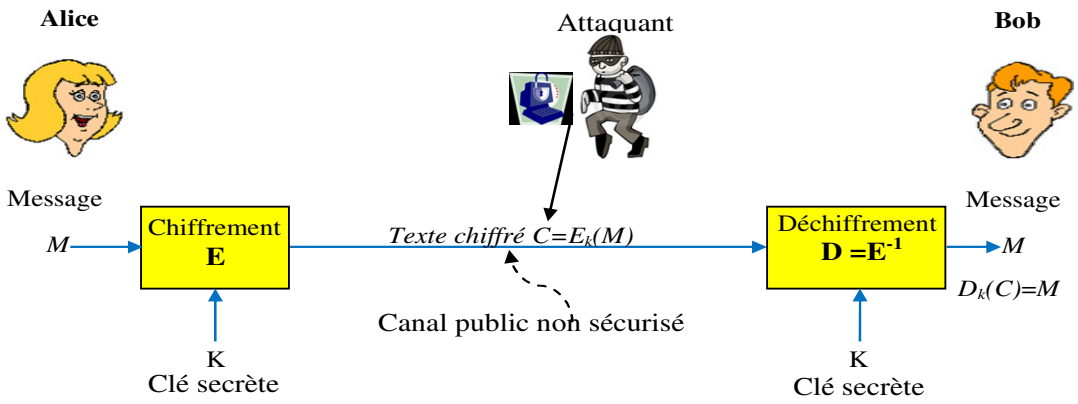

Figure 1.2. Schéma de principe d'un crypto-système

- Le chiffrement *symétrique* est défini par $D_k(E_k(m))=m$
- Le chiffrement *asymétrique* est défini par $D_d(E_e(m))=m$, où les clés e et d ne sont pas identiques, et, en plus d ne peut pas être dérivée facilement de e.

L'aléatoire est une autre notion fondamentale dans la cryptographie moderne. Un générateur de nombre pseudo-aléatoire (Pseudo Random Number Generator) PRNG est un dispositif capable de produire une séquence de nombres pseudo-aléatoires dont on ne peut pas facilement tirer des propriétés déterministes. Ces nombres peuvent être utilisés par des algorithmes de chiffrement pour masquer le texte en clair ou pour d'autres fins.

Un type de fonction aussi important est la *fonction à sens unique*. En général, une fonction a la propriété de sens unique [Menezes *et al.*, 1997] si :

- il est facile de calculer $f(x)$, si x est donnée, mais
- pour un y donné, il est impossible de trouver x.

1.2.2 Sécurité des systèmes avec des fonctions cryptographiques

L'utilisation de bonnes fonctions cryptographiques seules (algorithme de chiffrement symétrique ou asymétrique, fonction de hachage à sens unique, etc.) ne garantit pas la sécurité d'un système de communication. En effet, en plus des problèmes liés à la politique de la sécurité adoptée, il faut que la structure du système soit soigneusement conçue.

Dans la conception de la sécurité d'un système, lorsque deux entités veulent communiquer de manière sécurisée, il faut tenir compte du concept fondamental, qui stipule que toutes les fonctions cryptographiques doivent être publiques et seulement une composante de la paire des clés (e, d) ou k doit être gardée secrète. Ceci signifie que le système doit rester sécurisé même si les algorithmes et la structure du système sont mis à la disposition du public. Évidemment, les entités peuvent obtenir plus de sécurité si elles gardent en secret leurs algorithmes, mais cette mesure de la 'sécurité par l'obscurité' n'est pas réputée acceptable depuis 1970.

Afin d'être en mesure de communiquer d'une façon sécurisée, il faut d'abord partager les clés en toute sécurité. La gestion des clés secrètes est l'un des défis de la cryptographie (surtout dans le

cas du chiffrement symétrique) puisque la plupart des algorithmes de protection cryptographique s'appuient sur le concept d'une clé secrète et ces clés elles-mêmes doivent être protégées.

Dans les paragraphes suivants nous donnons un bref regard sur les différentes primitives cryptographiques qui peuvent être utilisées comme processus pour la construction des services de la sécurité énumérés dans le paragraphe 1.1.4. Les primitives les plus populaires sont : la primitive de chiffrement qui garantie le service de la « confidentialité », la primitive des codes d'authentification des messages (assuré par des fonctions de hachage) pour garantir le service de «l'intégrité », et les signatures numériques pour assurer la « non-répudiation ».

1.2.3 Chiffrement symétrique

La confidentialité des données est assurée soit par le chiffrement symétrique ou asymétrique. Dans le chiffrement symétrique les deux entités qui communiquent utilisent un algorithme de chiffrement/déchiffrement symétrique basé sur une même clé secrète *k*, comme montre la figure 1.3. Les algorithmes de chiffrement symétrique sont divisés en deux classes principales : algorithmes de chiffrement par bloc et algorithmes de chiffrement par flux.

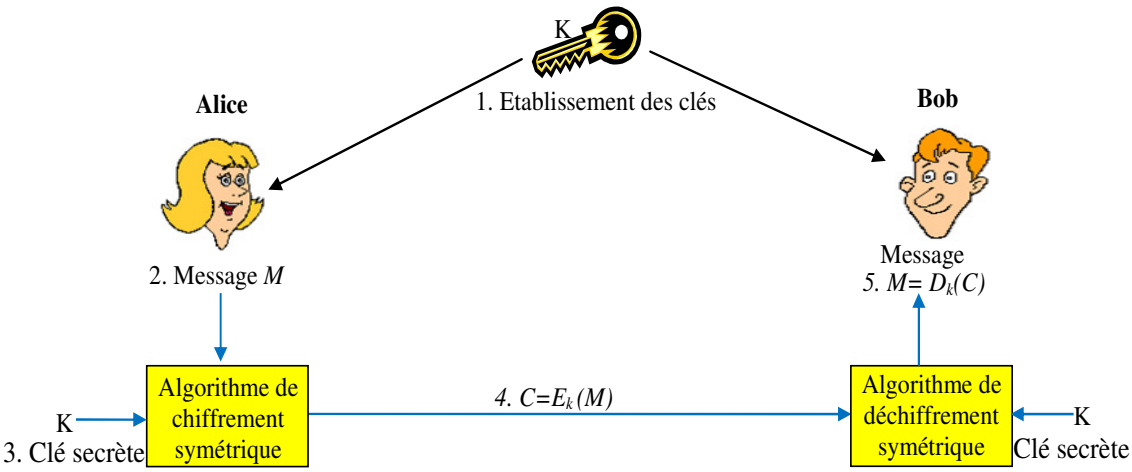

Figure 1.3. Chiffrement/déchiffrement symétrique avec une clé secrète

Dans le chiffrement par bloc le texte en clair se décompose en blocs de longueur fixe et les blocs seront chiffrés les uns après les autres en utilisant une clé (de taille fixe). Ainsi, pour n'importe quelle clé fixée, le chiffrement par bloc est une bijection: $C=E_k(M)$; $M = D_k(C)= D_k(E_k(m))=m$.

L'algorithme de chiffrement symétrique le plus connu et le plus robuste est l'AES (Advanced Encryption Standard) [FIPS 197, 2001]. Il utilise une longueur de bloc de 128 bits et une taille (minimale) de clé de : 128, 192, et 256 bits.

Les algorithmes de chiffrement par flux, traitent des données bit par bit ou mot par mot (un mot correspond à un octet ou un ensemble d'octets) par une technique simple, appelée *one-time pad*. A cet effet, un PRNG produit un masque binaire de taille égale au texte en clair pour effecteur le chiffrement. En outre, pour éviter la production du même masque (par l'utilisation d'une clé fixe), un compteur sera utilisé comme une entrée supplémentaire au générateur.

Les algorithmes de chiffrement par flux sont considérés moins robustes, mais plus rapides que les algorithmes de chiffrement par bloc.

Dans le cas du chiffrement symétrique, nos savons que les deux entités, Alice et Bob, avant d'échanger des messages en toute sécurité, doivent se mettre d'accord sur une clé secrète k. Ceci peut se faire suite à une rencontre physique entre Alice et Bob pour, mais cela n'est pas toujours possible. Donc, il se pose le problème important d'échange de clé en toute sécurité sans être interceptée. Une solution possible à ce problème est l'utilisation du chiffrement asymétrique que nous présentons au paragraphe 1.2.5.

1.2.4 Fonctions de hachage avec et sans clé

Une fonction de hachage h sans clé, est une primitive cryptographique sans clé et à sens unique. Elle possède les propriétés suivantes:

- *Compression*: $h(x)$ a une taille fixe (par exemple 256 bits) tandis que l'entrée x peut avoir une longueur quelconque plus grande;
- h (x) est facile à calculer.
- *Résistance à la seconde pré-image* : pour un x donné, il est infaisable de trouver un x_0 différent de x tel que $h(x) = h(x_0)$;
- *Résistance à la collision* : il est infaisable de trouver deux valeurs distinctes x et x_0 tel que $h(x) = h(x_0)$.

La fonction de hachage la plus utilisée est le SHA-256 (256 bits de sortie) qui appartient à la famille SHA-2 des fonctions de hachage approuvées par le [FIPS 180-4, 2012]. Une nouvelle fonction de hachage plus performante appelée KECCAK de la famille SHA-3 [NIST, 2013] est en cours de normalisation. Ces fonctions sont utilisées de différentes manières selon l'application. Un des cas important d'utilisation est de générer un condensât du message (appelé message digest en anglais), pour la signature électronique.

Une fonction de hachage avec clé est utilisée pour la génération des codes d'authentification des messages MAC (Message Authentication Code).

Il y a trois stratégies différentes dans la conception d'un code MAC : soit la conception directe, ou l'utilisation d'un chiffrement par bloc, ou l'utilisation des fonctions de hachage sans clé en tant que blocs de construction. La construction HMAC (Keyed-Hash Message Authentication Code) est un exemple de la troisième stratégie. Si k est la clé et x est l'entrée, alors la valeur MAC est obtenue par un double hachage:

$$\mathrm{MAC}(x) = \mathrm{HMAC}(x, k) = h((k \text{ xor } opad) \| h((k \text{ xor } ipad)\|x)), \qquad (1.1)$$

Où le symbole || est utilisé pour indiquer la concaténation, alors qu'*opad* et *ipad* sont des valeurs constantes utilisées pour des fins de bourrage. Comme on le voit, le processus HMAC mélange la clé secrète k aux données du message x, hache le résultat avec la fonction de hachage h, mélange à nouveau cette valeur de hachage à la clé secrète, puis applique la fonction de hachage une deuxième fois. Le résultat est souvent tronqué pour créer une valeur MAC courte (par exemple en extrayant les 32 premiers bits d'un total de 256 bits).

L'utilisation principale des codes MAC dans la sécurité de l'information est pour assurer l'intégrité d'un message : nous ajoutons un code MAC pour chaque message transmis à travers un

canal non sécurisé. Si l'entité réceptrice connaît la clé secrète k, elle peut alors calculer le MAC afin de vérifier que le message envoyé et le message reçu sont effectivement identiques.

1.2.5 Cryptographie à clé publique et ses composantes

Nous présentons dans ce paragraphe les notions de base et le principe de la cryptographie à clé publique (asymétrique). L'idée de chiffrement à clé publique est simple: nous utilisons des clés différentes pour le chiffrement et le déchiffrement, une publique et l'autre privée. Seule la clé publique e doit être disponible à tout le monde, tandis que la clé privée d doit être gardée secrète par son propriétaire. Ces deux clés sont reliées mathématiquement, et il est pratiquement infaisable de calculer la clé d à partir de la clé e.

Dans la figure 1.4, Alice veut envoyer à Bob un message sécurisé avec un algorithme de chiffrement asymétrique. Au début, Bob choisi une paire de clés (e_B, d_B) et envoie sa clé publique e_B à Alice. Alice chiffre le message M avec la clé e_B et l'envoie à Bob. Lorsque Bob reçoit le message chiffré, il utilise sa clé privée, d_B, pour le déchiffrer. Il est important de noter qu'il ne suffit pas que la clé de chiffrement e_B soit disponible au public, il faut en plus qu'elle soit authentifiée (garantie) par un organisme reconnu. Ceci est assuré par l'utilisation des *certificats*.

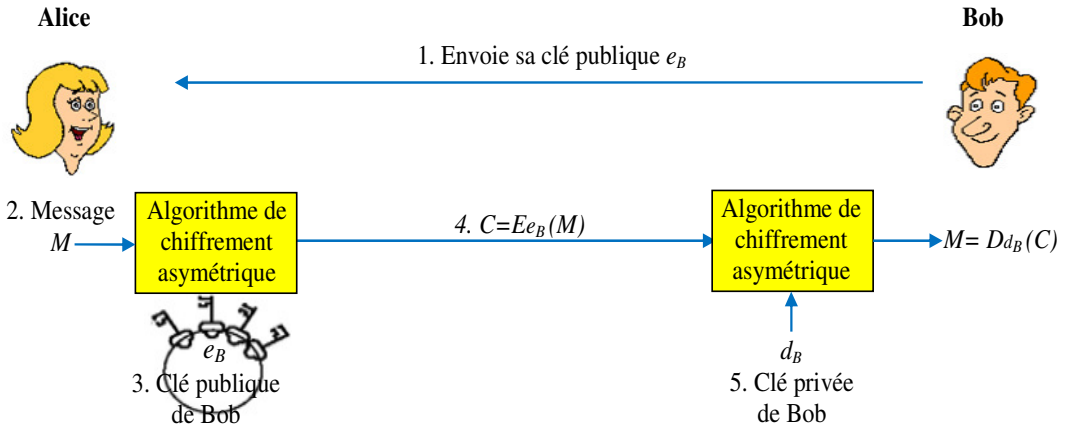

Figure 1.4. Chiffrement à clé publique et déchiffrement à clé privée

Avec le chiffrement à clé publique, le problème de la gestion et d'échange des clés symétriques d'une façon sécurisée est résolu facilement. Alice génère une clé symétrique aléatoire, elle le chiffre avec la clé publique de Bob et l'envoie à Bob. Même si ce message est intercepté, il ne peut pas être lu, car aucune personne à part Bob ne possède pas la bonne clé privée pour déchiffrer ce message. Bob déchiffre la clé symétrique avec sa clé privée. Maintenant les deux entités ont convenu sur une même clé secrète et peuvent commencer l'échange des messages en toute sécurité en utilisant le chiffrement symétrique.

Par ailleurs, la mise en œuvre du chiffrement asymétrique permet également de réaliser la *signature numérique* de document. Par exemple, Alice produit un document électronique et souhaite-le signer numériquement pour le protéger d'être modifié et prouver son authenticité. Pour cela, Alice calcule le condensât du document et le chiffre par sa clé privée d_A. Ensuite toute personne ayant accès à la clé publique d'Alice e_A peut ainsi vérifier la signature : en déchiffrant avec e_A le condensât, en calculant le condensât du document reçu, et en comparant les deux

valeurs obtenues. Si les valeurs sont identiques, la signature est validée, c.à.d. il est prouvé que le document n'a pas été modifié depuis qu'il a été signé, et c'est Alice qui l'a signé.

Le chiffrement asymétrique est au moins 100 fois plus lent que le chiffrement symétrique et il nécessite des clés plus longues (taille courante de l'ordre de 1024 bits). Il est principalement utilisé pour chiffrer des messages courts, comme par exemple la clé secrète d'un algorithme de chiffrement symétrique.

1.2.5.1 Certificats numériques

Pour qu'Alice puisse vérifier que la clé publique e_B, qu'elle a obtenue, appartient vraiment à Bob, on doit utiliser une infrastructure à clé publique PKI (Public Key Infrastructure) et plus précisément le *certificat numérique*. Ce dernier permet d'associer une clé publique à une entité (une personne, une machine, ...) afin d'en assurer la validité. Le certificat est en quelque sorte la carte d'identité de la clé publique, délivré par un organisme appelé autorité de certification (souvent notée CA pour Certification Authority).

Le certificat est un document signé par une autorité de confiance et contenant des informations d'identification de l'entité (comme son non, son adresse e-mail, l'employeur, etc.) groupées avec la clé publique de cette entité. Ce certificat est délivré et signé numériquement par un tiers de confiance (TTP) indépendant appelé l'autorité de certification (CA). Mais ceci suppose que le CA a déjà vérifié (souvent physiquement) l'identité de l'entité en question. Rappelons que l'entité peut être une personne ou une organisation.

Un certificat est essentiellement un véhicule pour le transport de clés publiques d'une façon vérifiable comme les certificats standard X.509 qui contient aussi le nom de l'émetteur du certificat (comme Verisign ou Thawte), des conditions de validité et d'autres attributs additionnels. Donc les certificats sont utilisés pour éviter les problèmes d'usurpation d'identité. Il y a des certificats personnels, délivrés à des personnes et souvent appelés identifiants numériques et d'autres délivrés à des organisations, implémentés dans le serveur de l'organisation. Le certificat doit être installé dans l'ordinateur ou dans un équipement de communication.

Parfois, la clé privée d'un utilisateur est compromise (par exemple volée), dans ce cas, le certificat doit être révoqué. Cela se fait souvent en incluant les certificats révoqués dans une liste des certificats révoqués (CRL, Certificate Revocation List) qui est gérée et signée par le CA.

1.2.5.2 Algorithmes de chiffrement et de signature asymétriques

La cryptographie à clé publique offre trois services essentiels qui sont :

- le chiffrement/déchiffrement qui assure la fonction de confidentialité (déjà présenté dans la figure 1.4) ;
- la création des signatures numériques qui assure l'authentification, l'intégrité et la fonction de non-répudiation (détail plus loin);
- l'échange des clés symétriques.

Ces services nécessitent l'utilisation d'algorithmes asymétriques : de chiffrement et de génération/vérification de la signature numérique.

Les principaux algorithmes de chiffrement asymétriques sont le RSA (Rivest Shamir Adleman) [Rivest *et al.*, 1978], El Gamal [El Gamal, 1984], et l'ECC (Elleptic Curve Cryptography) [Koblitz, 1987]. Le RSA est l'algorithme de chiffrement asymétrique le plus connu et le plus déployé parmi les algorithmes asymétriques. La robustesse du RSA est basée sur la difficulté pratique de factoriser un grand nombre entier n (qui est le produit de deux grands nombres premiers p et q) dans un temps raisonnable. Il utilise généralement des clés de 1024 bits. L'algorithme El Gamal est basé sur le calcul de logarithme discret et il a été conçu par Taher El Gamal en 1984. D'autres algorithmes basés sur les équations de calcul de circonférences des ellipses sont à l'origine de la cryptographie à courbe elliptique ECC. Bien que développé à partir des années 1985 et relativement robustes, l'usage de ce type de cryptographie reste marginal.

Les principaux algorithmes asymétriques utilisés pour la génération et la vérification d'une signature numérique sont le DSA (Digital Signature Algorithm), l'ECDSA (Elleptic Curve DSA) et le RSA [FIPS 186-3, 2009]. Ce dernier algorithme est adapté pour tous les usages (chiffrement/déchiffrement, génération de la signature, échange des clés). Dans les algorithmes DSA et ECDSA, un algorithme de chiffrement en plus sera nécessaire si on veut sécuriser les messages. Dans la figure 1.5, nous présentons le schéma de principe d'un crypto-système à clé publique qui assure la confidentialité et l'authenticité des messages (signer puis chiffrer).

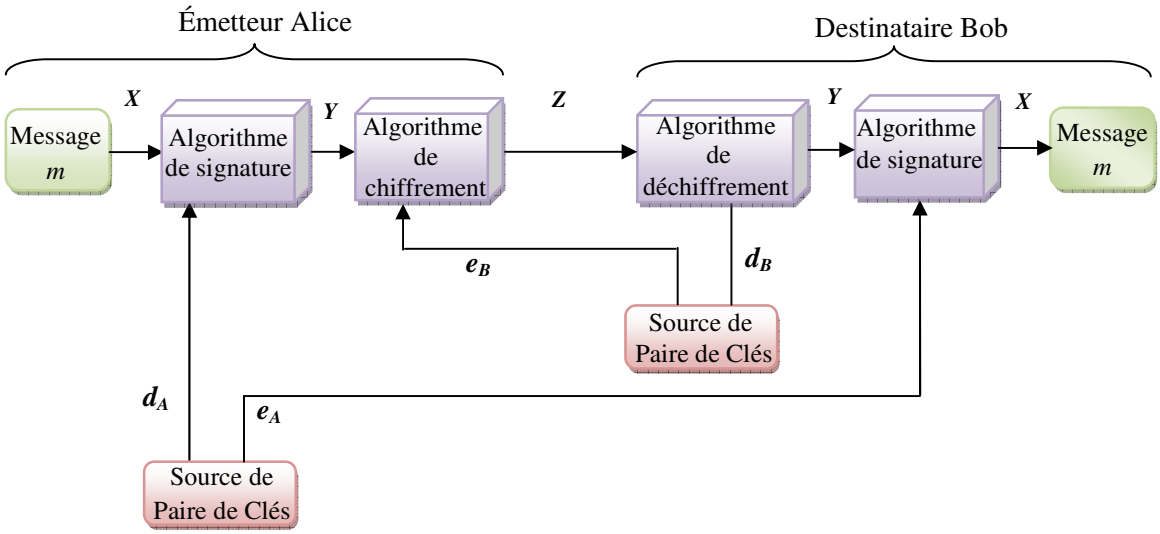

Figure 1.5. Crypto système à clé publique assurant la confidentialité et l'authenticité des messages

Dans cette figure, Alice veut envoyer un message signé et chiffré à Bob. Alice utilise sa clé privée d_A pour signer le message, et la clé publique de Bob e_B pour chiffrer le message signé. Lorsque Bob reçoit le message, il le déchiffre avec son clé privée d_B et vérifie la signature d'Alice par la clé publique e_A d'Alice. Dans le paragraphe suivant nous présentons le processus de génération et de la vérification de la signature effectué par Alice et Bob.

1.2.5.3 Signatures numériques

Les signatures digitales (ou numériques) sont utilisées pour assurer l'authentification du signataire, l'intégrité du message signé et la non-répudiation. Pour signer un message, le signataire utilise sa clé privée pour générer la signature et la signature sera vérifiée par la clé publique correspondante. Tout le monde peut vérifier la signature en utilisant la clé publique du signataire.

En général, pour générer une signature numérique, une fonction de hachage *h* à sens unique doit être appliquée sur le message M à signer afin d'obtenir une version condensée. Le condensât (*h*(M)) sera l'entrée d'un algorithme de signature numérique qui, avec l'utilisation de la clé privée du signataire (et parfois des informations supplémentaires en cas de DSA et ECDSA) permet de générer la signature numérique. La signature est ensuite attachée au message M pour être envoyée au destinataire. Ce dernier vérifie la signature en utilisant le même algorithme de signature numérique, la clé publique du signataire et la même fonction *h* qui a été utilisé en produisant la signature.

Dans la figure 1.6, Alice veut envoyer un message signé à Bob en utilisant par exemple l'algorithme RSA comme un algorithme de génération et de vérification de la signature. Après la génération du condensât, Alice utilise sa clé privée d_A pour chiffrer ce condensât. Le condensât chiffré est la signature digitale *s* qui sera attachée au message originale (M) pour être envoyée.

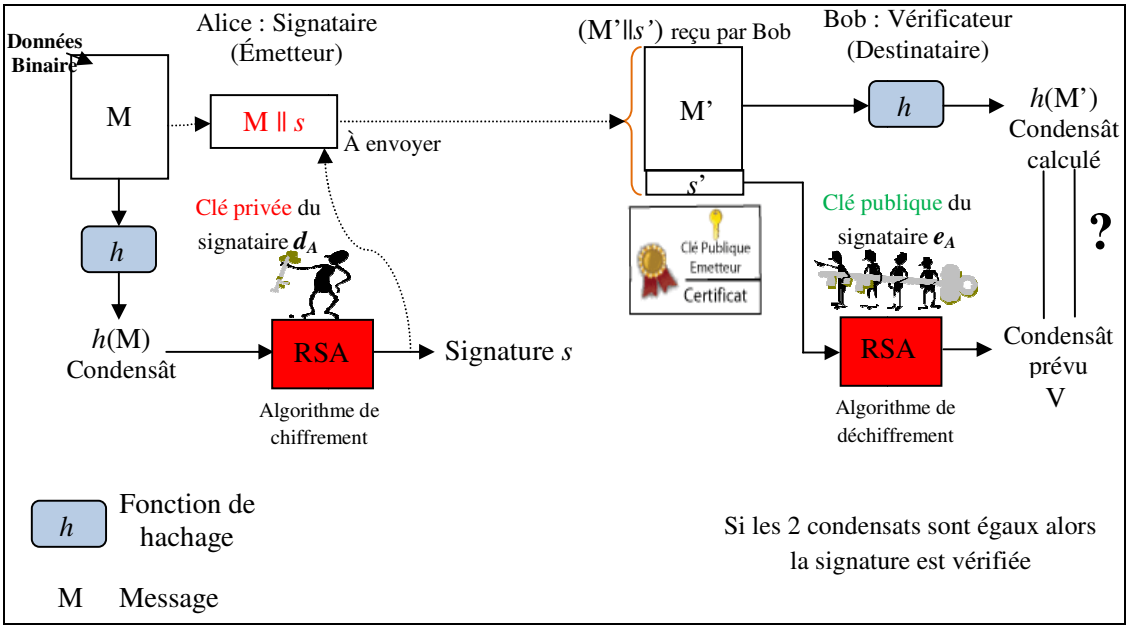

Figure 1.6. Processus de génération et de vérification de la signature en utilisant l'algorithme RSA

À la réception du message signé (M'‖s'), et afin de vérifier la signature et le contenu des données signées, Bob effectue les étapes suivantes :

1) il commence par le calcul du hachage du message reçu *h*(M'),
2) il déchiffre la signature *s*'avec la clé publique d'Alice e_A et l'algorithme RSA,

3) il compare la signature déchiffré (condensât prévu) avec le condensât qu'il a calculé.

Si les deux condensats sont égaux, Bob s'assure de l'intégrité du message reçu et confirme l'identité d'Alice car seule la clé publique d'Alice certifiée peut déchiffrer les données qui ont été chiffrées avec la clé privée d'Alice.

En plus, Bob peut utiliser la signature comme une preuve, pour démontrer à une partie-tierce (Third party) que la signature est générée par Alice. Ce qu'on appelle la non-répudiation, et Alice ne peut pas renier sa signature.

En utilisant les algorithmes DSA ou ECDSA, Alice doit utiliser en plus de la fonction de hachage h et sa clé privée e_A, un secret aléatoire k, qui se génère avec chaque message, et des paramètres publics (ou de domaine) KU_G. Ces derniers devront être envoyés à Bob avec la clé publique e_A. La signature produite de ces algorithmes est composée d'une paire (s et r) au lieu de s en RSA.

1.2.6 Cryptanalyse et attaques cryptographiques

Nous présentons dans ce paragraphe les concepts de base de la cryptanalyse. La cryptanalyse intègre l'ensemble des moyens permettant de déchiffrer un texte chiffré sans connaissance de la clé secrète, ou à partir de la découverte de cette dernière. En effet, l'attaquant, soit il analyse l'algorithme de chiffrement, soit il analyse les messages chiffrés et clairs, soit il essaye des clés secrètes. Les attaquants sont classifiés en deux types:

- Un attaquant passif (*passive attacker*): qui surveille (ou écoute) seulement la communication et essaie de casser la confidentialité.
- Un attaquant actif (*active attacker*): qui ajoute, supprime et modifie des messages. Il essaie de casser la confidentialité et aussi d'autres fonctions de sécurité.

Les modèles d'attaques cryptographiques (contre le chiffrement) peuvent être identifiés selon quatre niveaux de complexité décroissante :

- Attaque à texte chiffré seulement (*Ciphertext only attack*) : l'attaquant a connaissance seulement du texte chiffré (attaque la plus difficile) et essaye de trouver le texte en clair correspondant ou la clé secrète. C'est l'attaque à laquelle se trouvent régulièrement confrontés tous les services secrets.

- Attaque à texte en clair connu (*Known plaintext attack*) : l'attaquant détient le texte chiffré ainsi que le texte en clair correspondant et tente de trouver la clé de déchiffrement.

- Attaque à texte en clair choisi (*Chosen plaintext attack*) : l'attaquant peut choisir le texte en clair et obtient aussi le texte chiffré correspondant (et essaye à nouveau de trouver la clé de déchiffrement).

- Attaque à texte en clair choisi adaptative (*Adaptive chosen plaintext attack*) : c'est l'attaque la plus facile à mettre en œuvre. En effet, l'attaquant peut choisir les textes en clair qu'il donne à chiffrer au système et il peut les adapter (modifier) en fonction du résultat du chiffrement. Ceci permet au cryptanalyste (l'attaquant qui pratique la cryptanalyse) de modifier son texte en clair en fonction du résultat du chiffrement

correspondant et d'arriver ainsi assez vite à déchiffrer tout texte, si l'algorithme de chiffrement utilisé n'est pas assez robuste.

Parmi ces modèles, seulement les deux premières attaques sont disponibles pour un attaquant passif. Dans ce cas l'attaquant doit essayer toutes les combinaisons des clés possibles pour le déchiffrement. Cette attaque est connue sous le nom de l'attaque par force brute ou par recherche exhaustive. Par conséquent, la taille des clés secrète doit être assez importante pour que cette attaque ne soit pas techniquement faisable.

Ces modèles d'attaques s'appliquent également pour les attaques contre la protection de l'intégrité et de l'authentification.

1.3 Gestion des clés

La sécurité des communications repose en grande partie sur la confidentialité des clés secrètes. Ces clés secrètes qui sont de véritables données sensibles, nécessitent d'être gérées de manière fiable et confidentielle. La gestion des clés est définie comme étant l'ensemble des techniques et procédures qui ont pour but la distribution et l'établissement des clés secrètes entre deux ou plusieurs entités communicantes afin de réaliser les techniques cryptographiques.

Le domaine de gestion des clés est très important pour se protéger contre les attaques cryptographiques, pour cela il impose les requis suivants :

- *La durée de vie des clés secrètes* doit être telle qu'un adversaire n'ait pas assez de textes en clair et chiffré avec la même clé pour l'empêcher de déterminer la clé. La durée de vie d'une clé secrète dépend de son utilisation ;

- *La modalité de génération des clés* doit être telle qu'elle soit parfaitement aléatoire et valides (certaines algorithmes de chiffrement ont de mauvaises résultats avec certaines clés) ;

- *Le stockage des clés* doit être dans des zones sécurisées comme les cartes à puce qui sont résistantes à la manipulation ;

- *Les protocoles de distribution des clés* doivent être extrêmement robustes et ne doivent pas provoquer une grande charge sur les ressources du système.

Les clés sont exprimées en bits. L'attaque par force brute impose une longueur minimale pour les clés. Selon les recommandations d'Ecrypt [EcryptII, 2012] et [Barker et *al.*, 2007], la taille minimale de la clé symétrique pour une bonne sécurité des données doit être de 128 bits.

La gestion ou l'échange des clés d'une façon sécurisée constitue la phase la plus importante dans la sécurité de n'importe quel système de communication.

Nous traitons par la suite brièvement ce problème dans les systèmes de communications étudiés.

1.4 Chaos

La première personne qui a donné une définition claire au terme “chaos” [Chatterjee et Yilmaz, 1992] est le mathématicien “Henri Poincaré” en utilisant l’exemple des sphères. Il a affirmé que si on place une sphère réfléchissante sur laquelle on envoie un faisceau lumineux, la direction que prend le faisceau réfléchi dépend largement de la position d’origine. Avec deux sphères, la variation d’un dixième de degré dans l’angle de la source peut amener à une divergence de 180° entre les deux faisceaux. Cette sensibilité aux conditions initiales est la caractéristique propre de tout système chaotique.

1.4.1 Principe du crypto-système basée chaos

Plus tard, le chaos et suite a ses propriétés (que nous détaillons dans le paragraphe suivant) a été introduit dans le chiffrement des données. Les algorithmes de chiffrement chaotique utilisent des nombres pseudo-aléatoires générés par les fonctions (ou générateurs) chaotiques. Une fonction est dite chaotique, si elle est non linéaire et surtout si elle est sensible aux modifications, même extrêmement faibles de la valeur de la clé secrète qui est formée des conditions initiales et des paramètres du système. La séquence de nombre pseudo-aléatoire générée est utilisée par l’algorithme chaotique pour chiffrer le message en clair comme montre la figure 1.7.

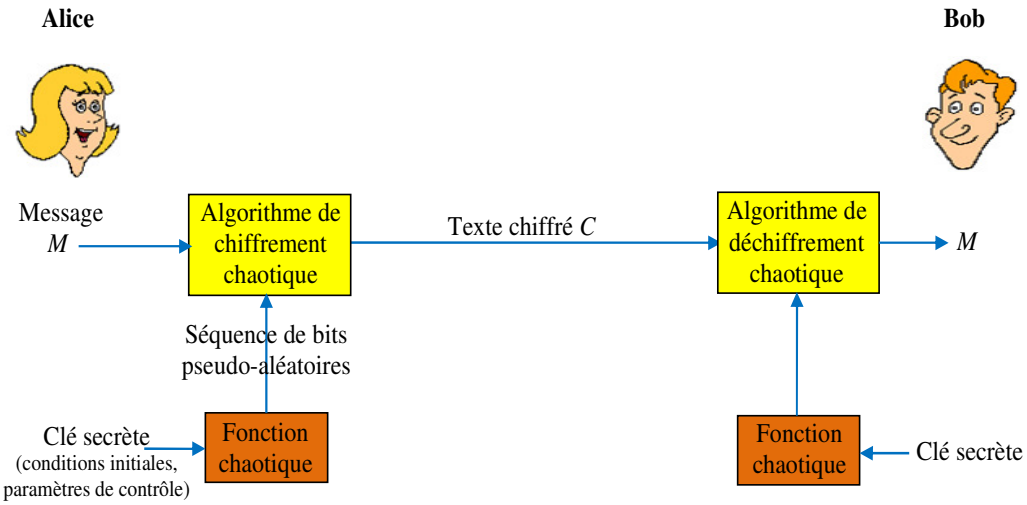

Figure 1.7. Schéma de principe d’un crypto-système basé chaos

À la réception, la même fonction chaotique est utilisée par Bob avec la même clé secrète pour générer la même séquence de nombres pseudo-aléatoires. Cette séquence sera utilisée par un algorithme de déchiffrement chaotique afin de récupérer le message en clair qui peut être des données numériques, une image, un texte, etc. Parmi les fonctions chaotiques, il y a la carte : logistique, PWLCM, Frey, et Skew tent map. Ces fonctions chaotiques sont des systèmes de récurrence. La caractéristique de l’hyper sensibilité à la clé secrète, est à l’origine de nombreux travaux de recherche scientifique, montrant l’apport des signaux chaotiques dans la sécurité des systèmes de communications.

La plupart des algorithmes de chiffrement/ déchiffrement basé chaos développés dans la littérature, sont des algorithmes à clé symétrique pour le chiffrement par bloc ou par flux. Parmi ces algorithmes on a le CBCSTI [Awad *et al.*, 2010] et d'autres proposé par [Bakhache *et al.*, 2011-a] et [El Assad, 2012].

1.4.2 Propriétés cryptographiques et chaotiques

La similarité entre les propriétés des fonctions chaotiques et les propriétés que nous trouvons dans les systèmes cryptographiques ont conduit au développement des crypto-systèmes basé chaos comme celui présenté dans la figure 1.7. Nous allons citer les principaux requis cryptographiques ainsi que les propriétés des fonctions chaotiques afin de montrer la correspondance entre les deux.

Les besoins cryptographiques essentiels sont:

1. *Sensibilité aux clés* : un changement d'un bit de la clé génère un texte chiffré totalement différent pour le chiffrement d'un texte en clair identique.
2. *Sensibilité au texte en clair* : un changement d'un bit de texte en clair change totalement le texte chiffré, même si la même clé est utilisée.
3. *Texte chiffré aléatoire* : le texte chiffré doit avoir un fort caractère aléatoire.

Les propriétés des fonctions chaotiques correspondantes aux besoins précédents sont :

1. *Sensibilité aux paramètres* : une petite variation des paramètres de contrôle génère deux trajectoires chaotiques très différentes même si elles partent de la même condition initiale.
2. *Sensibilité aux conditions initiales* : deux systèmes chaotiques qui partent des conditions initiales qui différent de très peu auront des trajectoires très différentes.
3. *Ergodicité* : les trajectoires qui partent des points arbitraires ont une distribution uniforme.
4. *Dynamique et déterministe* : avec un comportement apériodique pour les systèmes dynamiques à temps continu et périodique pour les systèmes à temps discrets.

Comme on voit, la correspondance est claire entre les trois besoins cryptographiques et les trois premières propriétés chaotiques.

1.4.3 Fonctions chaotiques numériques

Suite aux travaux de Pecora et Carroll des années 1990, le chaos a été implémenté avec succès dans les systèmes de communications analogiques dans le cas d'un canal idéal sans bruit.

L'implémentation numérique du chaos a posé le problème de la dégradation dynamique à cause de la précision finie de la représentation numérique. Quand les systèmes chaotiques sont discrétisés dans le temps, ses valeurs sont présentées avec une certaine précision finie *N*. Ainsi, des erreurs insurmontables de quantification seront introduites dans les itérations discrètes, et elles deviennent inévitablement cycliques [Li *et al.*, 2005] et leurs fonctions de distribution et de corrélation se détériorent. La période du cycle ainsi obtenu est parfois largement plus petite que le nombre total des états de la représentation finie. La figure 1.8 montre une description schématique d'une orbite typique d'un système chaotique numérique.

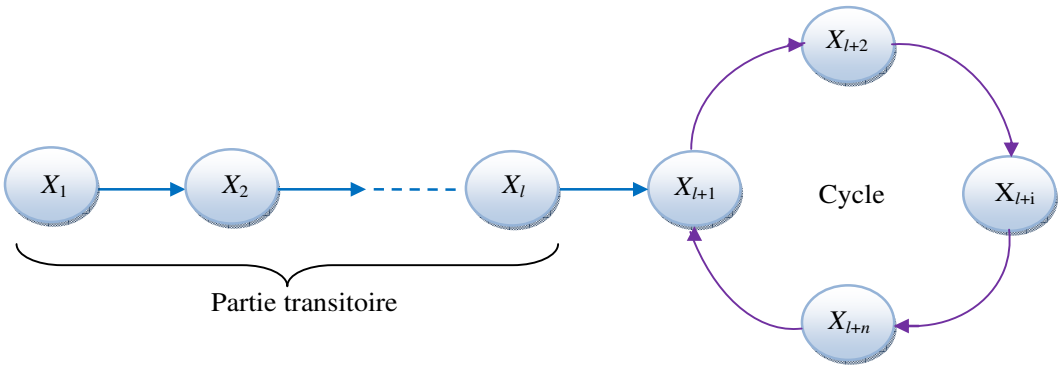

Figure 1.8. Pseudo-orbite d'un système chaotique numérique

En général, l'orbite chaotique est composée de deux parties reliées : une partie transitoire (branche) formée de : X_1, X_2. ….X_l, et une partie récurrente (ou cycle) formée de : X_{l+1}, X_{l+2},…, X_{l+n} où l est la longueur du régime transitoire, n la période du cycle, et $l+n$ est la longueur de l'orbite. Notons que si l=0, l'orbite est un simple cycle et que si n=0, l'orbite converge vers un point unique X_l.

1.4.4 Solutions pour éviter les effets de la précision finie N

Pour éliminer l'effet de la dégradation dynamique provoqué par la représentation finie, quatre solutions pratiques ont été proposées pour résoudre ce problème:

- *Utilisation d'une plus grande précision* : elle est fixé à N=32 bits pour maximiser la période de l'orbite et elle apporte des améliorations importantes ;
- *La perturbation aléatoire du système chaotique* : un perturbateur comme le générateur LFSR (Linear Feedback Shift Register) est introduit pour perturber la sortie du système chaotique numérique ;
- *La cascade de plusieurs systèmes chaotiques* : un autre système chaotique doit être ajouté pour augmenter la période de l'orbite chaotique ;
- Augmentation du délai dans le système.

Toutes ces solutions sont discutées et employées dans la technologie, et l'algorithme de perturbation a longuement été investi. La figure 1.9, montre le principe de perturbation et le cascade de deux systèmes chaotiques.

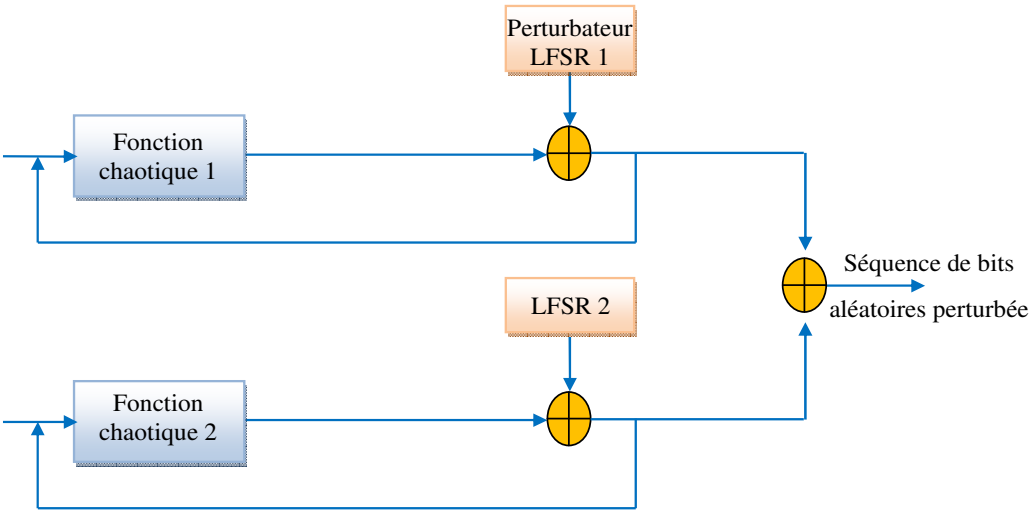

Figure 1.9. Cascade de deux fonctions chaotiques avec perturbation

1.5 Quelques définitions

Pour une meilleure compréhension nous allons enfin présenter quelques notions utilisées dans la thèse :

- *Protocole* : un protocole définit le format et l'ordre des messages échangés entre deux entités (ou plus) ainsi que les étapes (ou actions) que chaque entité doit suivre pour atteindre un certain objectif.
- *Nonce cryptographique* : un nombre à usage unique qui varie dans le temps. Il est utilisé dans les mécanismes cryptographiques.
- *Crypto période* : période de temps pendant laquelle une clé peut être utilisée.
- *Compromission* : divulgation ou obtention de données sensibles.

Dans les communications satellitaires, trois types de communications sont disponibles:

- *Unicast*: le message est adressé à un seul utilisateur. C'est le cas de la navigation IP par satellite.
- *Broadcast ou diffusion*: le message est adressé à tous les récepteurs situés dans la zone de couverture du satellite. C'est le cas de la diffusion de programmes de télévision par satellite. Bien sûr, ces communications n'ont pas besoin d'être sécurisées.
- *Multicast*: le message est adressé à un groupe d'utilisateurs. C'est le cas de la télévision payante et d'autres services IP par satellite, comme les vidéo conférences.

2. Sécurité et transfert efficace des communications IP Multicast à travers le DVB-S

2. Système de sécurité basé chaos pour les communications IP multicast à travers le satellite DVB

2.1 Introduction

L'intégration des communications satellitaire dans les réseaux IP d'aujourd'hui est un résultat direct des nouvelles tendances dans les télécommunications mondiales, où l'internet est utilisé par plus que 34.3% de la population mondiale [Internet world stats, 2013]. IP multicast est au centre de l'intérêt des activités internet pour les applications orientées groupes comme la vidéo conférence. Les communications IP multicast à travers le satellite offre des avantages importants tels que la grande couverture géographique, le déploiement rapide, et l'accessibilité dans les endroits isolés où aucune connectivité terrestre n'existe.

Le satellite géostationnaire GEO (Geostationary Earth Orbit) est visible à un tiers de la terre, et grâce à sa capacité de diffusion n'importe qui peut recevoir ce que le satellite envoie dans sa zone de couverture. Ceci est très utile (pratique) lorsque les satellites sont utilisés pour les applications telles que la diffusion de télévision (TV broadcasting), mais il peut poser des problèmes pour les applications multicast, où le message est destiné à un groupe restreint d'utilisateurs dispersés géographiquement. Par conséquent, les paquets multicast sont envoyés par le satellite exclusivement aux faisceaux ponctuels (spot beams) contenant des membres. Cela a de nombreux avantages allant de l'amélioration de la sécurité à l'utilisation efficace de la bande passante descendante. Cependant, n'importe quel utilisateur qui se trouve à l'intérieur de la couverture d'un spot beam peut recevoir et accéder aux données multicast transmises. Afin de contourner cet inconvénient, des solutions existent pour assurer la sécurité et surtout la confidentialité des données transmises, mais ne sont pas très performantes [Cruickshank *et al.*, 2008], [Duquerroy *et al.*, 2004], [Pillai et HU, 2006].

Le DVB-S [ETSI 301 192, 2004] fait partie de la famille des standards DVB. Il a été initialement proposé pour la diffusion de l'audio numérique et de la télévision. Plus tard, quelques méthodes d'encapsulation ont été proposées pour permettre les liaisons IP sur DVB. La transmission des paquets IP à travers le DVB-S utilise le MPEG-2 TS (MPEG-2 Transport Stream) pour le multiplexage et le transport des trames de données de longueur fixe (188 octets) sur la liaison satellite. Dans ce contexte, se pose la question de l'encapsulation et la segmentation efficace des paquets IP multicast, afin de permettre au processeur du satellite (OBP, On-Board Processor) de commuter les segments MPEG-2 (Moving Pictures Expert Group) reçus aux endroits (spot beams) appropriés.

Dans la littérature, il y a deux approches de commutation permettant la commutation des segments MPEG-2 à bord du satellite. Ces approches sont le 'label-switching' et le 'self-switching' [Filali *et al.*, 2004]. Les méthodes d'encapsulation existantes, l'ULE (Unidirectional Lightweight Encapsualtion) et le MPE (Multi protocol Encaspulation) ne permettent pas d'offrir les informations nécessaires au satellite pour effectuer la commutation à son bord. D'où la nécessité d'adapter ces méthodes ou la conception d'une nouvelle méthode pour opérer avec les approches existantes afin de bénéficier des technologies OBS (On Board Switching) et spot beam.

Pour assurer la sécurité dans un groupe multicast, un système de gestion de clés doit être utilisé pour distribuer une clé de groupe à tous les membres du groupe pour le chiffrement des données.

La gestion des clés est un processus assez compliqué et coûteux (consomme de la bande passante et de la puissance de traitement) à cause du problème de renouvellement des clés (rekeying). Ceci, est la cause principale qui limite l'évolutivité du groupe et les ressources du réseau, surtout lorsque le nombre des membres du groupe multicast est grand et lorsque ces membres sont très dynamiques (fréquence élevée de 'joindre/quitter') [Hubenko *et al.*, 2007]. En effet, pour maintenir un bon niveau de sécurité lorsqu'un nouveau membre se joint à un groupe ou lorsqu'un membre existant quitte un groupe, la clé du groupe doit être mise à jour et redistribuée avec un grand nombre de clés à tous les membres autorisés. Donc, il est primordial de minimiser le coût du trafic provenant de la gestion des clés. Plusieurs techniques de gestion de clés ont été proposées dans la littérature [Rafaeli et Hutchison, 2003]. Parmi ces techniques, il a été prouvé que le LKH (Logical Key Hierarchy) est le système de gestion de clés le mieux adapté aux transmissions multicast satellitaire [Howarth *et al.*, 2004].

D'un autre côté, l'étude et l'apport du chaos ont suscité beaucoup d'intérêt par les chercheurs dans plusieurs domaines scientifiques incluant le domaine des télécommunications. En effet, les caractéristiques importantes de chaos telles que les bonnes propriétés cryptographiques, la très haute sensibilité aux conditions initiales et le comportement dynamique non linéaire des cartes chaotiques, incitent à leur utilisation dans des crypto-systèmes ou dans de nouveaux systèmes de communication pour la sécurité des données. Donc, nous proposons d'utiliser les séquences chaotiques pour améliorer la sécurité des transmissions multicast sur DVB.

Dans ce chapitre, nous proposons un nouveau système de sécurité pour les communications IP multicast à travers le DVB-S. Il s'appuie d'abord sur une nouvelle méthode d'encapsulation appelée Enhanced ULE (EULE) dérivée de la méthode standard ULE pour assurer un transfert efficace de données multicast via le satellite GEO. L'EULE peut utiliser l'une de deux approches citées plus haut pour la commutation des segments MPEG-2. Puis, pour sécuriser les données multicast (trames EULE), nous proposons un mécanisme de sécurité performant qui utilise une entête d'extension. Ensuite, pour résoudre le problème de renouvellement fréquent de clés, le système de sécurité proposé utilise un système de gestion de clés, appelé TLKH (Two-Tiered LKH), basé sur une architecture à deux couches de distribution des clés LKH indépendantes : une couche satellitaire et une couche terrestre. Dans les deux couches, pour plus de sécurité, les clés sont générées par des séquences chaotiques et sont transmises dans des paquets particuliers définis à cet effet. Par ailleurs, le chiffrement des données et des clés est assuré par des algorithmes de chiffrement basé chaos.

Dans la première partie de ce chapitre, nous présentons le contexte de notre étude : la transmission des données IP multicast à travers le DVB-S et ses exigences de la sécurité ainsi que la gestion des clés et les approches de commutation existantes. Puis, nous décrivons le système de sécurité multicast proposé qui intègre de nouveaux mécanismes d'encapsulation, de sécurité et de gestion de clés. L'analyse et les avantages du système de gestion de clés proposé sont ensuite détaillés. Ensuite, nous évaluons la performance du système de sécurité proposé en termes de nombre de messages de rekeying et de la consommation de la bande passante, et nous montrons que les résultats obtenus sont meilleurs que ceux obtenus par les solutions existantes actuellement. Enfin nous concluons ce chapitre.

2.2 Communications IP Multicast à travers le DVB-S

Le standard DVB-S a été conçu pour la diffusion de la radio et de la télévision. Ultérieurement, il est devenu possible de l'utiliser pour permettre la transmission des paquets IP via le satellite. Dans ce paragraphe, nous exposons d'abord l'architecture générale du système DVB-S et la transmission des données IP multicast à travers ce système. Puis, nous présentons les approches de commutation existantes au niveau satellitaire (label-switching et self-switching) et les exigences pour opérer avec elles. Nous abordons ensuite les critères de la sécurité pour ces communications et les solutions existantes de la littérature. Enfin nous présentons le système de gestion des clés LKH qui est le meilleur système pour ce type de communication.

2.2.1 Architecture du système DVB-S

L'architecture générale du système qui permet l'accès à l'internet via le satellite géostationnaire (GEO) en utilisant DVB-S est représentée dans la figure 2.1. Les deux entités principales qui assurent le transfert des données et la connexion avec le satellite GEO sont: le NCC (Network Control Center) et le RCST (Return Channel via Satellite Terminal). Le NCC est le noyau du réseau satellitaire, il contrôle tous les RCST du réseau. Le RCST est le terminal terrestre qui assure une liaison bidirectionnelle avec le satellite GEO.

En général, différents canaux de retour peuvent être utilisés par les utilisateurs selon les disponibilités offertes, par exemple via: ISDN (Integrated Services Digital Network), GPRS (General Packet Radio Service), satellite, etc. Les RCST utilisent le canal de retour par satellite DVB-RCS (Return Channel via Satellite) élaboré en 1999 pour faciliter la communication réciproque et pour éviter l'augmentation du trafic sur les réseaux terrestres qui souvent aboutit à la réduction du QoS (Quality of service) ou à son blocage. En effet, les canaux aller-retour (DVB-S/RCS) permettent un meilleur contrôle de la QoS et de la gestion du réseau par les opérateurs.

Le fournisseur de service Internet (ISP, Internet Service Provider), transmet les paquets IP à ses clients (via les RCST) en utilisant la liaison satellitaire. Les paquets IP doivent être encapsulés et transportés par le flux des segments MPEG-2 (MPEG-2 Transport Stream) au satellite. Ce dernier achemine les données MPEG-2 en s'appuyant sur les deux technologies : le 'Spot beam' et l'OBS (On-Board Switching).

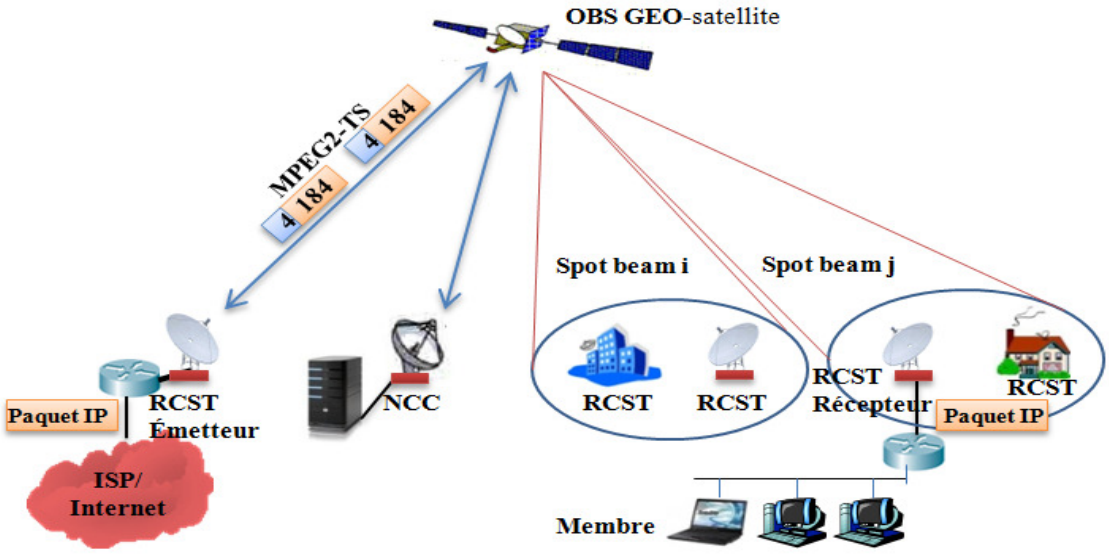

Figure 2.1. Architecture du système DVB-S supportée par les technologies

'Spot beam' et 'OBS'

La technologie 'Spot beam' ou 'faisceau ponctuel' consiste à diviser le faisceau global du satellite en un certain nombre de faisceaux étroits (spot beam). Ceci présente deux avantages importants :

1) les besoins en puissance des terminaux terrestres sont réduits, ce qui permet l'utilisation d'antennes de dimensions plus petites et donc de faible coût ;

2) la réutilisation fréquentielle (réutiliser les mêmes fréquences) dans des faisceaux différents augmente la capacité du secteur spatial. Ajoutons aussi que le nombre des personnes qui peuvent recevoir des données qui ne leur sont pas destinées, est considérablement réduit, ce qui augmente la sécurité essentiellement contre les menaces passives (écoute).

D'un autre côté, l'OBS fournit un procédé de commutation et de routage des données à bord du satellite. Le processeur on-board (OBP) commute les segments MPEG-2 reçus aux ports de destination (spots) correspondants.

2.2.2 Structure du RCST émetteur et transmission des paquets IP via DVB-S

Un ensemble d'opérations doivent être effectuées sur chacun des paquets IP afin de les préparer à être transmis à travers le DVB-S. Dans la figure 2.2, nous montrons un exemple de la procédure de transmission des paquets IP par le RCST émetteur du fournisseur d'accès (ISP).

Le routeur ISP reçoit les paquets IP entrants destinés aux utilisateurs du réseau satellitaire (de l'internet) et les transmet à l'encapsulateur IP qui forme la liaison entre les équipements classiques du réseau et les équipements de communications DVB-S. L'encapsulateur applique une méthode d'encapsulation, implémentée dans sa couche d'encapsulation, sur chaque paquet IP. L'encapsulation consiste à ajouter un entête et un en-queue sur chaque paquet IP pour former la trame SNDU (Sub-Network Data Unit). Ensuite il emballe la trame générée (SNDU) dans un ensemble des segments MPEG-2 de taille fixe égale à 188 octets comme indiqué dans la figure 2.2. Chaque segment MPEG-2 possède 4 octets d'en-tête et 184 octets de payload.

Pour l'opération de transmission, le champ le plus important de l'entête MPEG-2 est l'identificateur de paquet (PID, Packet Identifier). C'est un champ de 13 bits utilisé pour indiquer à quel flux de données (ou paquet) le payload du segment appartient. Après cela, le train binaire des segments MPEG-2 doit être modulé (par le modem) et transmis (par l'antenne de RCST) au satellite GEO avec le niveau de puissance nécessaire et dans la bande de fréquence allouée.

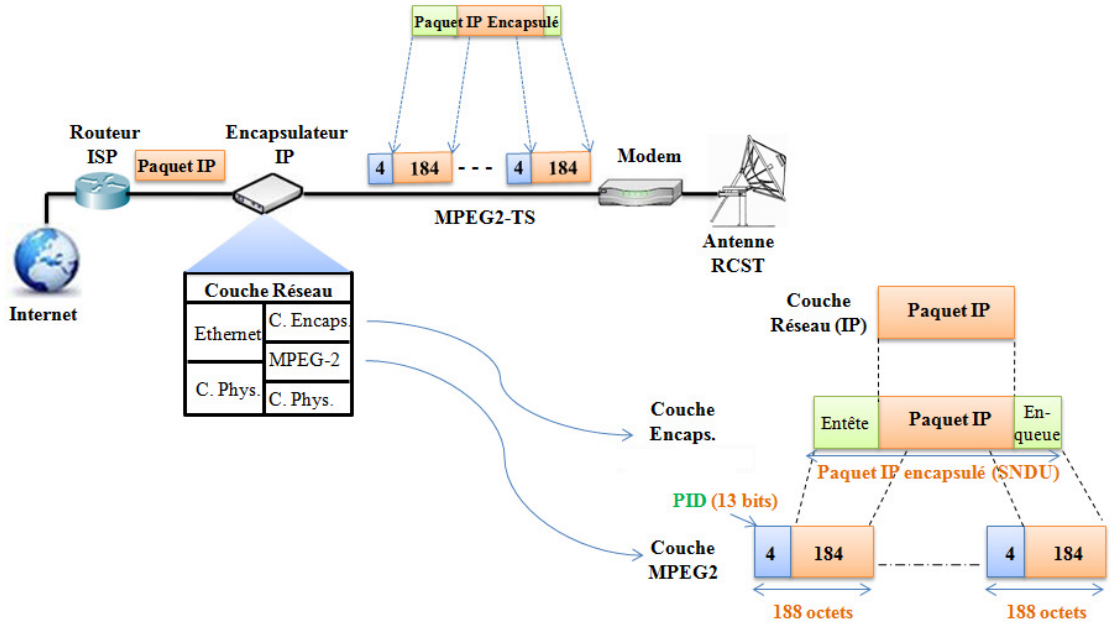

Figure 2.2. Structure du RCST émetteur d'un ISP et transport des paquets dans des segments MPEG-2

Dans la figure 2.2 le MPEG-2 TS contient un flux de données IP (IP-stream), mais en général il peut contenir d'autres types de flux, comme l'audio et la vidéo. Cela peut être fait en multiplexant les différents types de flux multimédia par un multiplexeur MPEG-2.

Le RCST récepteur doit être équipé avec: une antenne RCST qui reçoit le signal satellitaire, un modem pour la démodulation et la reconstruction de flux MPEG-2, un récupérateur des paquets IP (IPPRU-IP Packet Recovery Unit) pour extraire les paquets IP à partir des segments MPEG-2, et un routeur pour acheminer les paquets vers l'utilisateur final qui peut être un ordinateur ou un réseau LAN (Local Area Network).

Les RCST acquièrent toutes les informations nécessaires pour la transmission de données à partir du NCC. Ces informations comprend le PID qui sera utilisé pour envoyer les données, ainsi que le PID réservé pour les messages de contrôle (CTRL/MNGM PID) [Filali *et al.*, 2004].

2.2.2.1 Méthodes d'encapsulation existantes : MPE, IP-Optimized scheme, ULE

L'encapsulation des paquets IP est nécessaire pour que ces paquets puissent être transportés par le DVB-S dans les segments MPEG-2. Parmi les méthodes d'encapsulation existantes, il y a deux méthodes standards disponibles qui sont : le "MPE" et le "ULE", et une méthode spécifique, le "IP-Optimized scheme" proposée pour les communications multicast [Filali *et al.*, 2004]. Les

méthodes standards ajoutent un entête et un en-queue à chaque paquet PDU (Protocol Data Unit) pour former la trame SNDU et faciliter son transport. La méthode "IP-Otimized scheme" se caractérise par l'ajout seulement d'un entête comme montre la figure 2.3.

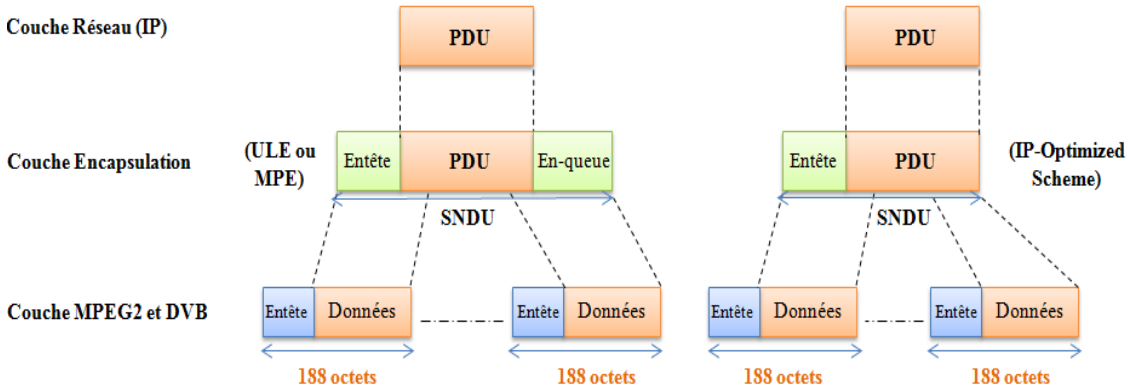

Figure 2.3. Encapsulation MPE, ULE et IP-Optimized des paquets IP et leurs transports dans des segments MPEG-2

Chaque SNDU est ensuite transportée avec une ou plusieurs segments MPEG-2. Dans la figure ci-dessus, nous avons utilisé l'appellation plus générale PDU pour désigner les paquets IP, car les encapsulations présentées peuvent aussi être utilisées pour d'autres protocoles au niveau réseau, et non seulement pour la couche IP.

Il a été démontré par [Collini-Nocker et Fairhurst, 2004] et [Hong *et al.*, 2005] que la méthode d'encapsulation ULE est la meilleure dans le DVB-S/DVB-RCS. Elle présente plusieurs avantages par rapport à la méthode MPE, tels que : le support natif de nouveaux protocoles, un coût de traitement inférieur et une meilleure efficacité de transport.

La méthode IP-optimized a les inconvénients suivants : elle n'utilise pas un en-queue qui assure la protection de tout le paquet contre les erreurs de transport, et elle ne peut pas être utilisée pour les communications unicast ce qui la rend limitée et inefficace. Pour cela dans notre travail nous nous intéressons seulement à la méthode d'encapsulation ULE qui est la meilleure à ce jour.

2.2.2.2 Encapsulation ULE et extension d'entête

L'encapsulation ULE crée la trame de données SNDU en ajoutant un 'entête' de 10 octets et un 'en-queue' de 4 octets au PDU de la couche réseau. La trame ULE est présentée dans la figure 2.4.

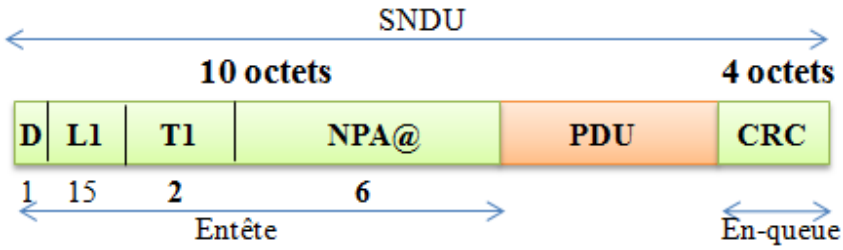

Figure 2.4. Trame de données ULE

L'entête est constitué de quatre champs qui sont:

- ✓ Le champ *D* : composé d'un seul bit (0 ou 1) indiquant la présence ou l'absence du champ facultatif *NPA@* (Network Point of Attachment address).
- ✓ Le champ longueur *L1* : composé de 15 bits, indiquant la longueur de la trame SNDU en octets, à partir du premier octet qui suit le champ *T1* jusqu'au fin dernier octet de l'en-queue.
- ✓ Le champ Type *T1* : formé de 16 bits, indiquant le type de PDU porté par le SNDU ou le type d'extension de l'en-tête s'il existe. Dans le premier cas, T1 prend une valeur supérieure à 0x0600 (par exemple *T1*=0x0800 si le PDU est un paquet IP), et dans le deuxième cas il prend une valeur inferieure à 0x0600.
- ✓ Le champ facultatif NPA@: formé de 6 octets, contient l'adresse de destination de la trame SNDU. Habituellement, il s'agit d'une adresse MAC (Medium Access Control) similaire à celui utilisé par le protocole Ethernet.

L'en-queue est un code CRC-32 (Cyclic Redundancy Check) appliqué sur tout le SNDU.

Si on veut ajouter des services supplémentaires, telle que la sécurité, on a besoin d'ajouter de l'information additionnelle. L'encapsulation ULE possède un mécanisme très souple qui permet l'extension de l'en-tête original comme c'est montré sur la figure 2.5.

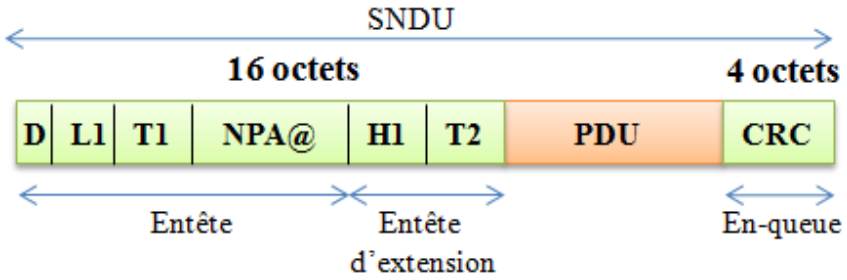

Figure 2.5. Encapsulation ULE avec une extension de son entête

Le champ *H1* est l'entête d'extension et sa structure est déterminée par le champ type *T1*. La structure de champ *T1* dans ce cas est représentée dans la figure 2.6. Il existe des entêtes d'extension prédéfinies [Fairhurst et Collini-Nocker, 2005] et des valeurs non assignés pour des types d'entête d'extension qui pourrait être utilisés pour les nouvelles entêtes d'extension. Le champ *T2* indique le type de PDU qui est en cours, similaire au champ *T1* de l'ULE sans l'en-tête d'extension.

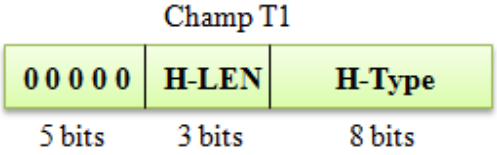

Figure 2.6. Structure du champ *T1* pour indiquer un entête d'extension ULE

Dans la figure 2.6 les 5 bits les plus signifiants sont mis à 0 pour signaler qu'il s'agit du type d'un entête d'extension et pas du type du PDU transporté. La partie *H-LEN* du champ *T1* (sur 3 bits) indique la longueur de l'entête d'extension. La partie *H-Type* indique le type de l'entête

d'extension. Les valeurs que peut prendre ce champ sont attribuées par l'organisation IANA (Internet Assigned Numbers Authority).

2.2.2.3 Structure du segment de transport MPEG-2

Le standard MPEG-2 permet le transport des données IP (trames ULE), des données vidéo ou des données audio utilisées par le DVB et par d'autres systèmes de communications. Il est utilisé par tous les systèmes DVB-S et par une partie des systèmes DVB-S2, car le DVB-S2 peut aussi utiliser le MPEG-4 ou le GSE (Generic Stream Encapsulation), selon le type d'implémentation. Le format de multiplexage des données MPEG-2 TS, utilise des segments de transport ayant une longueur fixe de 188 octets présenté dans la figure 2.7.

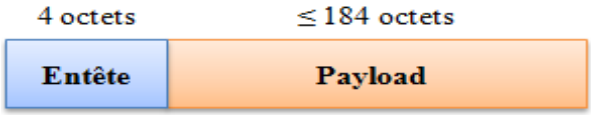

Figure 2.7. Segment de transport MPEG-2

La longueur de 188 octets a été choisie pour permettre l'interopérabilité avec l'ATM (Asynchronous Transfer Mode). Chaque segment MPEG-2 peut être transporté par 4 cellules ATM, chacune de taille fixe égale à 53 octets. Aussi, deux segments MPEG-2 peuvent être transportés par 8 cellules ATM si l'AAL5 (ATM Adaptation Layer 5) est utilisé (d'après la recommandation de l'ITU-T I.363.5). Tous les segments contiennent un entête de 4 octets. La structure de l'entête est présentée dans la figure 2.8.

Mot de synchronisation 0x47	Indicateur erreur	**PUSI**	Indicateur de priorité	**PID**	Contrôle Brouillage	Champ Adaptation	Compteur de continuité
1 octet	1 bit	1 bit	1 bit	13 bits	2 bits	2 bits	4 bits

Figure 2.8. Entête du segment MEPG-2

Les champs les plus importants de l'entête sont :

- Mot de synchronisation : utilisé pour établir la synchronisation du récepteur ou la resynchronisation en cas d'erreur.
- Indicateur erreur : indique s'il y a des erreurs dans le segment MPEG-2 TS.
- PUSI (Payload Unit Start Indicator) : indique si le segment MPEG-2 contient le début d'une nouvelle trame SNDU, c.à.d. un nouveau paquet IP encapsulé.
- PID (Packet Identifier) : indique le flux de données ou programme (télévision) auquel la trame ULE ou le paquet appartient.

2.2.3 Approches de commutation au niveau satellitaire (Label-Switching et Self Switching)

La commutation des segments MPEG-2 au bord du satellite se fait grâce à l'une des deux approches appelées : 'label-switching' et 'self-switching', [Filali *et al.*, 2004]. Dans l'approche 'label-switching', la commutation est basée sur deux tables de commutation principales

maintenues par le satellite. Ces tables sont mises à jour périodiquement, l'une par le NCC et l'autre par le satellite lui-même. Cette approche utilise cinq tables qui sont partagées dans les différentes composantes du réseau : une table de mappage (mapping table) au niveau du RCST émetteur, deux tables de commutation sur le satellite (une temporaire et une permanente) et deux tables au niveau du RCST récepteur (une pour l'abonnement et l'autre pour le filtrage).

Dans la deuxième approche 'self-switching', la commutation est basée sur un label (4 octets ou plus) inclus dans chaque segment. Ce label est ajouté par le RCST émetteur à l'entête de chaque segment MPEG-2. Cette approche utilise trois tables : une table de mappage (mapping table) au niveau du RCST émetteur contenant un nouveau champ (par rapport au label-switching) pour représenter le label et deux tables au niveau du RCST récepteur (une d'abonnement et l'autre de filtrage) similaires aux tables utilisées par le 'label-switching'.

Le fonctionnement de ces approches nécessite l'utilisation d'un ensemble des messages qui doivent être échangés entre le NCC et les RCST afin de mettre à jour les différentes tables sur les différentes entités et pour améliorer l'OBS. Ces messages sont traités par le protocole SMAP (Satellite Multicast Adaptation Protocol) détaillé dans [Filali *et al.*, 2004] et qui est une protocole de convergence de la signalisation entre les protocoles de routage multicast terrestres (PIM Sparse Mode, Distance Vector Multicast Routing Protocol,..) et les messages de signalisation DVB.

Le SMAP a définit un nouveau descripteur DVB (DVB-descriptor) appelé 'session-descriptor' qui permet au NCC d'annoncer et de diffuser toutes les informations relatives aux sessions multicast actives. Il indique également le format des messages : 'new-session' utilisé en cas de déclenchement d'une nouvelle session par un RCST, et le message 'join/leave' envoyé par un RCST au NCC lors d'un événement de 'joindre/quitter' un groupe.

Pour travailler avec les deux approches de commutation citées plus haut, deux conditions principales doivent être respectées :

1) fournir une manière pour identifier les différents flux de données multicast par d'autres moyens que l'identification par le PID.

2) le mécanisme proposé ne devrait pas surcharger les ressources satellitaires qui sont coûteuses.

2.2.4 Critères de la sécurité des communications IP par DVB satellitaire

Une analyse générale de la sécurité des données IP avec encapsulation ULE a été faite par [Cruickshank *et al.*, 2009]. Le cas particulier de l'analyse des menaces et des exigences de la sécurité des données IP à travers le satellite DVB, utilisant l'encapsulation ULE a été réalisé par [Iyengar *et al.*, 2007]. Pour ces types de communications, chaque nouveau système ou protocole qui propose une nouvelle solution de sécurité doit tenir compte des critères de sécurité décrits dans ces deux documents cités ci-dessus.

Nous présentons dans ce paragraphe, les types d'attaques auxquelles les données IP satellitaires sont exposées et les services de sécurité nécessaires pour les contrer. Notant qu'aucun des standards (MPEG-2, DVB-S, ULE) décrits jusqu'ici incluent des services de sécurité.

2.2.4.1 Attaques actives et passives

Comme la communication multicast à travers le DVB-S est une transmission sans fil, elle est donc vulnérable à diverses attaques qui sont divisées en deux catégories: actives et passives. Dans les attaques actives l'intrus peut par exemple injecter ses propres messages dans le flux binaire (bitstream) ou modifier les informations dans le bitstream. Les attaques actives connues sont :

- Le masquage (Masquerading): lorsqu'une entité prétend d'être une autre entité, comme l'encapsulateur IP qui implique l'accès au fournisseur.
- La modification des messages d'une manière non autorisée.
- Attaques de relecture (Replay attacks): Lorsque l'intrus transmet des exemplaires supplémentaires d'anciens messages au récepteur.
- Attaques de type déni de service, (Denial of Service attacks): Lorsqu'une entité ne parvient pas à exécuter sa propre fonction.

La réalisation de ces attaques par des internautes n'est pas très compliquée dans les systèmes de communication IP classiques. Mais, dans un système avec un environnement de diffusion (notre cas d'étude) l'implémentation de ces attaques est très difficile et nécessite des moyens très sophistiqués [Iyengar *et al.*, 2007]. En effet, les segments MPEG-2TS portés par le système DVB-S aux RCST, utilisent des codes FEC et une technique d'entrelacement de bits de plusieurs segments TS consécutives. En plus, l'antenne d'un RCST est très bien pointée vers le satellite relai et elle est accordée à la fréquence du fournisseur. C'est difficile à un intrus d'altérer les paquets IP originaux d'un certain flux de données ou d'insérer ses propres paquets.

Dans ce système la menace la plus répandue est la menace passive (écoute) car les terminaux de réception ne sont pas coûteux et la zone de couverture des satellites est assez vaste. Un attaquant peut surveiller facilement les transmissions multicast afin d'accéder soit aux données transmises soit à différentes informations relatives au trafic entre les RCST.

2.2.4.2 Exigences de la sécurité des données IP via DVB-S

Les services de sécurité qui ont été dérivées pour contrecarrer les attaques passives et actives sont les suivants :

- ✓ La confidentialité des données (Data confidentiality): est le service de sécurité le plus important contre les menaces passives, car aucun récepteur non autorisé ne peut accéder aux données en clair PDU.

- ✓ L'intégrité des données et l'authentification de la source (Data integrity and authentication): services permettant de contrer les menaces actives.

- ✓ Protection contre les attaques de relecture (Protection against replay attacks)*:* service atteint par l'utilisation des numéros de séquences.

- ✓ Authentification du terminal de la couche liaison de données (Link layer terminal authentication): il est nécessaire dans le cadre du protocole de la gestion des clés, et il est réalisé pendant l'échange initial de clés, avant d'établir une liaison sécurisée.

- ✓ Confidentialité en avant (Forward secrecy): est nécessaire pour empêcher un membre partant d'un groupe à accéder au trafic multicast futur.

- ✓ Confidentialité en arrière (Backward secrecy): est nécessaire pour empêcher un membre joignant un groupe à accéder au trafic multicast déjà envoyé (dans le passé).

La confidentialité en avant et en arrière est assurée par le système de gestion des clés qui assure une mise à jour de la clé de groupe (rekeying-renouvellement de clés) à chaque fois qu'un membre joint/quitte un groupe.

2.2.5 Solutions de la sécurité existantes

Afin de répondre aux exigences de la sécurité, différentes solutions de sécurité ont été proposés par [Duquerroy *et al.*, 2004], [Pillai et HU, 2006], [Cruickshank *et al.*, 2008], et [Caragata *et al.*, 2010]. Parmi ces solutions, il y a un ensemble qui utilise les entêtes d'extension ULE et d'autres qui utilisent l'IPsec (IP Security) en mode tunnel. Dans ce paragraphe nous présentons ces deux solutions et nous analysons leurs points faibles et forts. Cette analyse nous a permis de proposer un nouveau mécanisme plus performant appelé SEULE.

2.2.5.1 Utilisation des entêtes d'extension de l'ULE

On a vu que le format d'encapsulation ULE permet l'utilisation d'une ou plusieurs entêtes d'extension, si des services supplémentaires sont nécessaires. Les solutions par l'extension de l'entête en vu de la sécurité ont été proposés par [Cruickshank *et al.*, 2008] et [Caragata *et al.*, 2010]. Ces extensions tiennent compte des critères de la sécurité et du format de l'entête d'extension présentés précédemment. La sécurité de l'encapsulation ULE est réalisée au niveau de la couche liaison de données. Elle renforce (et non remplace) les autres mécanismes de sécurité tels que l'IPsec (qui sera détaillé dans l'Annexe A) ou le TLS (Transport Layer Security) qui sont réalisés aux couches supérieures. Le format de l'une de ces extensions d'entête est montré dans la figure 2.9.

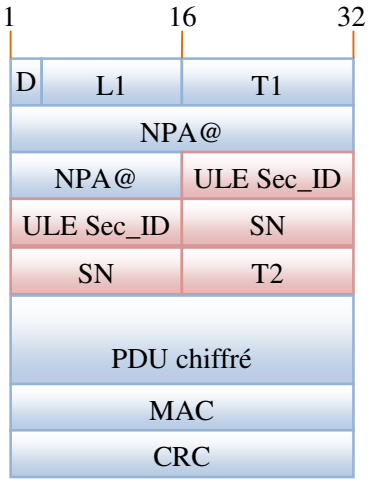

Figure 2.9. Entête d'extension de l'ULE pour la sécurité

Cet entête d'extension est inspiré d'IPSec et il contient trois champs :

- *ULE Sec-ID* (ULE Security Identifier): est un champ qui indique l'association de sécurité de manière similaire que le SPI (Security Parameter Index) d'IPSec.

- *SN* (Sequence Number): est un compteur qui s'incrémente pour chaque nouveau SNDU, permettant ainsi d'empêcher les attaques de relecture des messages.
- *T2* : indique le type de PDU encapsulé.

Les services de sécurité fournit par cet extension d'entête sont:

- ✓ *Confidentialité de données*: service réalisé par le chiffrement des PDU.
- ✓ *Authentification de l'origine des données*: service réalisé par un code d'authentification des messages, le MAC (Message Authentication Code).
- ✓ *Intégrité des données*: service réalisé aussi par le MAC, qui permet à l'utilisateur de vérifier que le message n'a pas été modifié tout au long de sa transmission.
- ✓ *Protection contre l'envoie multiple*: chaque SNDU a dans son entête d'extension, un numéro de séquence SN différent. Ceci permet au récepteur de ne traiter que les SNDU qui ont un numéro de séquence valide, ce qui empêche l'attaquant de réutiliser un ancien SNDU.

Cette solution a deux inconvénients: le premier vient du fait que la solution proposée s'inspire trop d'IPSec, qui a été conçu pour les réseaux IP terrestres. Le deuxième inconvénient, tient du fait que la solution en question est incomplète, car elle ne traite réellement pas le problème de la gestion des clés qui est une question très importante pour tout système de sécurité. En effet, la solution en question recommande tout simplement l'utilisation des processus de gestion des clés utilisés par IPSec. Cependant, il est clair que ces techniques de gestion des clés ne sont pas adaptées pour les systèmes satellitaires.

2.2.5.2 Utilisation d'IPSec en mode tunnel

La deuxième solution qui permet d'offrir la sécurité aux communications IP par DVB satellitaire, est l'établissement d'un réseau virtuel privé entre le fournisseur des services IP (ISP) et le client. La solution la plus répandue est l'utilisation d'IPSec en mode tunnel.

Ceci implique des inconvénients tels qu'un taux des données ajoutées fort (jusqu'au 20% pour IPSec) et un retard important pour l'établissement des clés. Le retard important est dû au fait que l'établissement des clés est réalisé après que le fournisseur et le client aient échangés au moins deux messages, ce qui prend entre 1 à 1.5 secondes dans le cas des communications satellitaires.

2.2.6 Gestion des clés pour les communications multicast satellitaire

Pour maintenir la sécurité dans les systèmes de communication multicast, un certain nombre de systèmes de gestion de clés ont été proposés [Rafaeli et Hutchison, 2003]. Parmi ces systèmes on cite: le système Flat, LKH, Iolus, Kronos, etc. Il a été prouvé que le système hiérarchique des clés LKH (Logical Key Hierarchy) est le système de gestion de clés le plus adapté aux communications multicast satellitaire et peut gérer avec succès des grands groupes dynamiques [Howarth *et al.*, 2004]. Pour cela nous présentons dans ce paragraphe le système LKH avec son système de base flat.

2.2.6.1 Systèmes de gestion des clés Flat et LKH

Dans tous les systèmes de communication multicast, il y a toujours un contrôleur de la clé du groupe GKC (Group Key Controller) qui gère les communications et qui est responsable de la

génération et de la distribution des clés. Dans un groupe donné, on suppose qu'il y a *N* membres GM (Group Member) qui ont accès à l'information et à la clé du groupe GK (Group Key). Bien sûr, pour notre cas, le GKC est géré par le fournisseur des services IP et les GM sont les utilisateurs. Chaque GM a sa propre clé secrète appelé UK (Unique Key) qui est partagé avec le GKC. Tous les GM partagent une clé commune avec le GKC, qui est la clé GK, avec laquelle le chiffrement/déchiffrement du trafic multicast est effectué à l'intérieur du groupe.

Dans le protocole flat, qui est l'approche la plus simple, chaque membre est directement connecté au GKC et ne dispose que d'une seule clé UK comme montre la figure 2.10. Ainsi, lorsqu'une mise à jour de la clé GK survient, la nouvelle clé GK est alors envoyée à chaque membre, un par un, chiffrée par l'UK de chaque membre. Donc, le coût de rekeying (renouvellement de clés) et de stockage du protocole flat est *N*. Par exemple, dans la figure 2.10 avec *N*=15 membres et un 16ème membre qui se joint au groupe, le GKC transmet 16 messages contenant chacun la nouvelle clé GK, chiffrée avec la clé UK(i), i=1,…16.

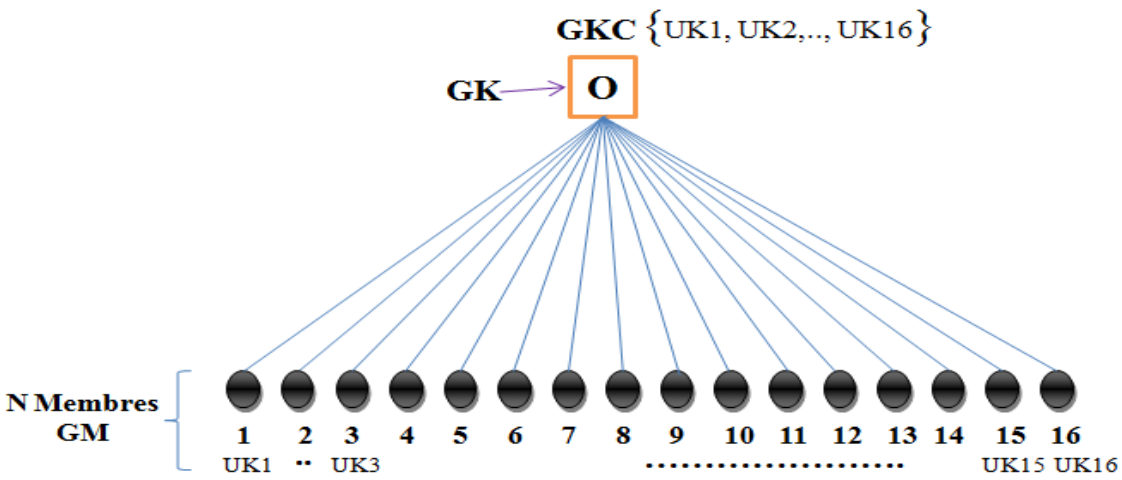

Figure 2.10. Système de gestion de clés Flat

Le protocole LKH conçu pour le traitement des grands groupes utilise une structure arborescente pour réduire le coût de rekeying. La Figure 2.11 montre un exemple de LKH avec quatre niveaux de clés (les cercles représentent les membres GM, numérotés de 1 à 16). Le GKC génère tout l'arbre des clés {A, B,…,O} et chaque membre aura connaissance des clés le concernant, à savoir les clés de chiffrement KEK (Key Encryption Keys) qui forment une liaison entre son UK et la GK. Par exemple, le GM11 détient le vecteur des clés {UK11, les KEK *F, K, N* et la GK *O*}. Si le membre en question (le GM11) veut quitter son groupe multicast (rekey event), le GKC doit changer toutes les clés {*F, K, N, O*} connues par GM11.

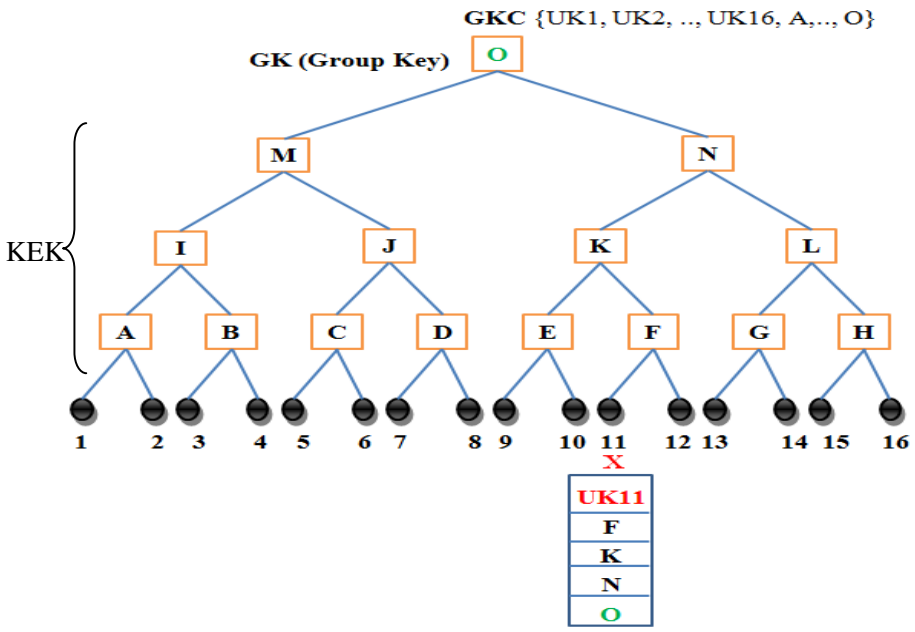

Figure 2.11. Structure de l'arbre des clés LKH

Le changement se fait de bas en haut, c.à.d. le GKC commence par la génération d'une nouvelle valeur pour la clé *F* et la transmet au GM12, chiffrée avec UK12. Ensuite une nouvelle valeur de la clé *K* est générée et est envoyée chiffrée avec E aux membres GM9 et GM10, et avec la nouvelle clé F au GM12. Puis une nouvelle valeur pour *N* est générée et est envoyée chiffrée avec la clé K aux membres GM9, GM10, et GM12, et avec la clé L aux membres GM13 à GM16. Enfin une nouvelle valeur pour *O* est générée et est envoyée chiffrée avec M aux GM1à GM8, et avec la nouvelle clé N aux GM9, GM10, GM12 à GM16.

En général, $klog_k(N) - 1$ transmissions à travers le satellite GEO sont nécessaires pour faire cette modification de clés afin d'assurer la confidentialité en avant (forward secrecy), où k est le facteur de branchement de l'arbre (dans notre exemple k=2, et 7 est le nombre de transmissions) [Howarth *et al.*, 2004]. D'une façon similaire, $klog_k(N)$ transmissions sont effectuées lorsqu'un nouveau membre rejoint le groupe afin d'assurer la confidentialité en arrière (backward secrecy). Donc, le coût de changement dans l'appartenance au groupe (in group membership) est très élevé, pour cela nous proposons ci-dessous un nouveau système pour le réduire.

2.3 Système de sécurité multicast proposé

Dans ce paragraphe nous décrivons en détails, la solution que nous proposons pour assurer la sécurité et le déploiement efficace des communications IP multicast à travers le satellite DVB. Notre solution travaille au niveau ULE et intègre une large variété de composantes. En effet, nous proposons une variante améliorée du standard ULE appelé EULE (Enhanced ULE), comprenant : un entête d'extension à l'EULE pour la sécurité, un système de gestion des clés, un transport des données relatives à la gestion et à la synchronisation des clés et une structure modifiée de l'encapsulateur/décapsulateur ULE.

Nous montrons que toutes ces composantes sont compatibles et ensemble, elles forment une solution de transfert des données IP multicast sécurisé très robuste. Nous décrivons ensuite, tous les détails nécessaires en vu d'une implémentation pratique du système proposé.

2.3.1 Présentation générale du système

Nous proposons un nouveau système de sécurité multicast basé chaos qui s'appuie sur:

- une méthode d'encapsulation ULE améliorée, notée EULE qui satisfait les exigences de fonctionnement avec les approches de commutation 'label-switching' et 'self-switching' et assure un transfert efficace de multicast via le satellite DVB sans l'ajout d'aucun overhead.

- un mécanisme de sécurité qui tient compte des critères de la sécurité décrits précédemment dans le paragraphe exigences de sécurité et des caractéristiques des communications satellitaires. Ce mécanisme est conçu pour sécuriser les trames EULE (SEULE) par l'utilisation : d'un entête d'extension, d'un code MAC, et par le chiffrement des PDU. Le processus de chiffrement est réalisé avec des algorithmes basés sur des fonctions chaotiques.

- un système évolutif de gestion de clés à deux couches LKH indépendantes (TLKH) afin de réduire le coût de renouvellement des clés à travers la liaison satellite. Dans les deux couches, et pour plus de sécurité, les clés sont générées par un générateur de séquences chaotiques.

- un nouveau type de paquet appelé KPDU (dont nous définissons le format) pour le transport des clés et des paramètres de sécurité. Ces paquets sont envoyés par le GKC et reçus par les RCST. Le RCST envoie un message au GKC à propos des clés (message d'alarme), seulement en cas de pertes de la synchronisation des clés.

Enfin nous exposons les modifications nécessaires sur l'encapsulateur/décapsulateur d'un RCST afin d'implémenter le système de sécurité proposé.

2.3.2 Méthode d'encapsulation proposée EULE

Après l'encapsulation d'un paquet IP par l'ULE, le paquet encapsulé (trame ULE) est segmenté en plusieurs segments MPEG-2 transmis vers le satellite. Ce dernier reçoit les segments de données MPEG-2 et les traite à son bord afin de les acheminer aux ports convenables. Par conséquent, chaque segment MPEG doit contenir une information supplémentaire qui aide l'OBP à gérer la commutation. Comme le standard ULE ne prévoit pas ce genre d'information, nous proposons une amélioration de la méthode d'encapsulation ULE, appelée EULE permettant d'inclure l'information nécessaire pour gérer la commutation. Cette modification est faite dans l'entête ULE, elle permet de prendre en considération les exigences des transmissions multicast tout en étant en cohérence avec les approches de commutation.

Cette modification consiste à remplacer le champ NPA@ par trois nouveaux champs indiqués dans la figure 2.12 et qui sont :

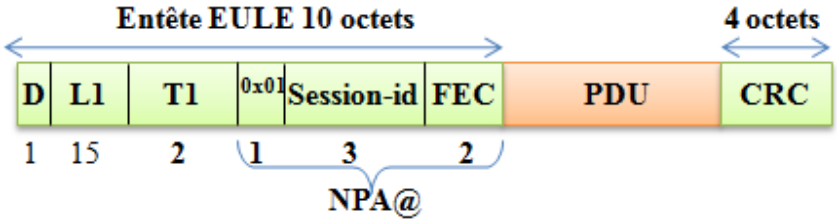

Figure 2.12. Encapsulation EULE proposée

- Le champ ***Adapt*** : composé d'un octet constant ***=0x01*** qui signifie qu'il s'agit d'une adresse ou d'une trame multicast et non d'une trame unicast [Fairhurst et Collini-Nocker, 2005].

- Le champ ***Session-id*** : un identifiant de 3 octets, dont le rôle est d'identifier une session multicast et par suite un groupe multicast. La valeur du ***Session-id*** est attribuée et diffusée par le NCC après une demande par le RCST émetteur qui annonce une nouvelle session 'new-session'. La taille de ce champ donne la capacité d'avoir 2^{24} sessions simultanément.

- Le champ ***FEC*** (Forward Error Correction) : c'est un champ de 2 octets, dont le rôle est de garantir une transmission fiable de l'entête améliorée.

Il est à noter que dans notre proposition [Ahmad *et al*., 2012-a], nous n'avons ajouté aucun bit supplémentaire au standard ULE (spécifiquement à l'entête). En outre, l'EULE fournit les informations requises (session-id) pour opérer avec les approches de commutation qui rendent plus efficace l'acheminement et le filtrage des données. La méthode proposée possède les propriétés suivantes: adaptée à la fois aux communications unicast et multicast, facile à mettre en œuvre et bénéficie des avantages de l'ULE tels que la simplicité, l'efficacité et la flexibilité

2.3.3 Mécanisme proposé pour la sécurité des trames EULE (SEULE)

Nous proposons un mécanisme de sécurité pour les communications multicast qui répond aux exigences de la sécurité des communications satellitaire. Le mécanisme proposé (SEULE), qui travaille au niveau de la couche liaison de données, assure la sécurité des trames EULE (Secured EULE, SEULE) et améliore tous les services de sécurité. À cet effet, nous proposons un nouveau entête d'extension qui contient deux champs : le champ *PN* (Packet Number) de 32 bits, et le champ *T2* de 16 bits indiqués dans la figure 2.13 [Ahmad *et al*., 2011].

Le PN est un nombre qui croit de façon monotone indiquant le numéro de la trame transmise. Ce champ, inspiré du champ *SN* d'IPSec, offre une protection contre les attaques de relecture des messages et est utilisé comme un 'nonce' pour la dérivation d'une nouvelle clé de chiffrement pour chaque PDU. Le champ T2 désigne le type du PDU encapsulé. Le type du nouvel entête d'extension est défini par le champ T1 et sa valeur doit être assignée par l'IANA.

Pour assurer l'intégrité et l'authenticité des données, nous proposons de remplacer le CRC de la trame EULE par un code MAC (Message Authentication Code) comme indiqué dans la figure 2.13. Le MAC protégera le PDU, et également tout l'entête de l'EULE sécurisé (SEULE), offrant ainsi une protection contre une large gamme d'attaques actives. Pour apporter la confidentialité

des données, tous les PDU transmis doivent être chiffrés. Nous proposons l'utilisation d'un système de dérivation des clés qui permet le chiffrement de chaque PDU par une clé transitoire TK (le détail sur ce point sera vu plus tard).

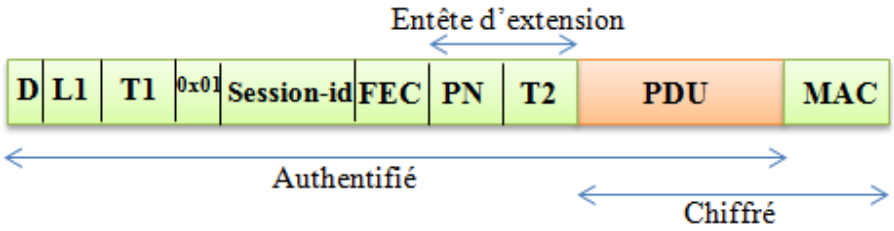

Figure 2.13. Trame EULE sécurisée (SEULE)

Pour le chiffrement, nous recommandons d'utiliser l'un des algorithmes de chiffrement chaotiques des références suivantes [El Assad, 2012], [Farajallah *et al.*, 2013], [Awad *et al.*, 2010], [Bakhache *et al.*, 2011-a] et [Bakhache *et al.*, 2011-b] ou d'autres algorithmes publics qui sont considérés robustes du point de vue cryptographique, comme l'AES (Advanced Encryption Standard) [FIPS 197, 2001]. Les algorithmes chaotiques énumérés ont été initialement proposés pour le chiffrement d'images, mais ils peuvent facilement être modifiés pour chiffrer n'importe quel flux de données. Pour renforcer la sécurité, le chiffrement ne couvre pas uniquement le PDU, mais aussi le code MAC.

La procédure d'authentification du terminal au niveau de la couche liaison de données est réalisée par le système de gestion de clés proposé TLKH. Elle est basée sur une clé secrète pré-partagée entre le fournisseur et les RCST par d'autres moyens de communication que la liaison satellitaire (par exemple, par une carte à puce).

2.3.4 Interfonctionnement entre l'encapsulation SEULE et les approches de commutation

L'interfonctionnement entre l'encapsulation SEULE et l'une des approches de commutation permet : d'assure un support efficace et sécurisé d'IP multicast ; de faciliter la transmission des paquets multicast, et de garantir également aux RCST récepteurs de filtrer les paquets entrants afin de réduire le trafic inutile. Ci-dessous, nous exposons ces aspects pour chacune des approches.

2.3.4.1 Encapsulation SEULE avec label-switching

Suite à notre proposition, le RCST émetteur doit encapsuler et sécuriser le trafic multicast provenant d'une source multicast (encapsulation SEULE) pour être ensuite envoyé au satellite GEO. Dans le cas de label-switching, ces opérations sont décrites dans la figure 2.14.

Le RCST émetteur doit se référer à sa table de mappage pour assurer la correspondance entre la paire des adresses IP (@ IP source, @ IP multicast) et sa session-id correspondant. Ainsi, le RCST émetteur obtient la session-id et l'insère dans l'entête EULE (égal à 0x000001 dans cet exemple) puis ajoute les deux autres champs (*Adapt* et *FEC*). Ensuite, il applique la sécurité sur la trame EULE obtenue. Il authentifie et chiffre cette trame par des clés d'intégrité et de chiffrement

dérivées de TK. Enfin, il partage la trame SEULE sur plusieurs segments MPEG-2 (TS) qui seront envoyés successivement vers le satellite GEO.

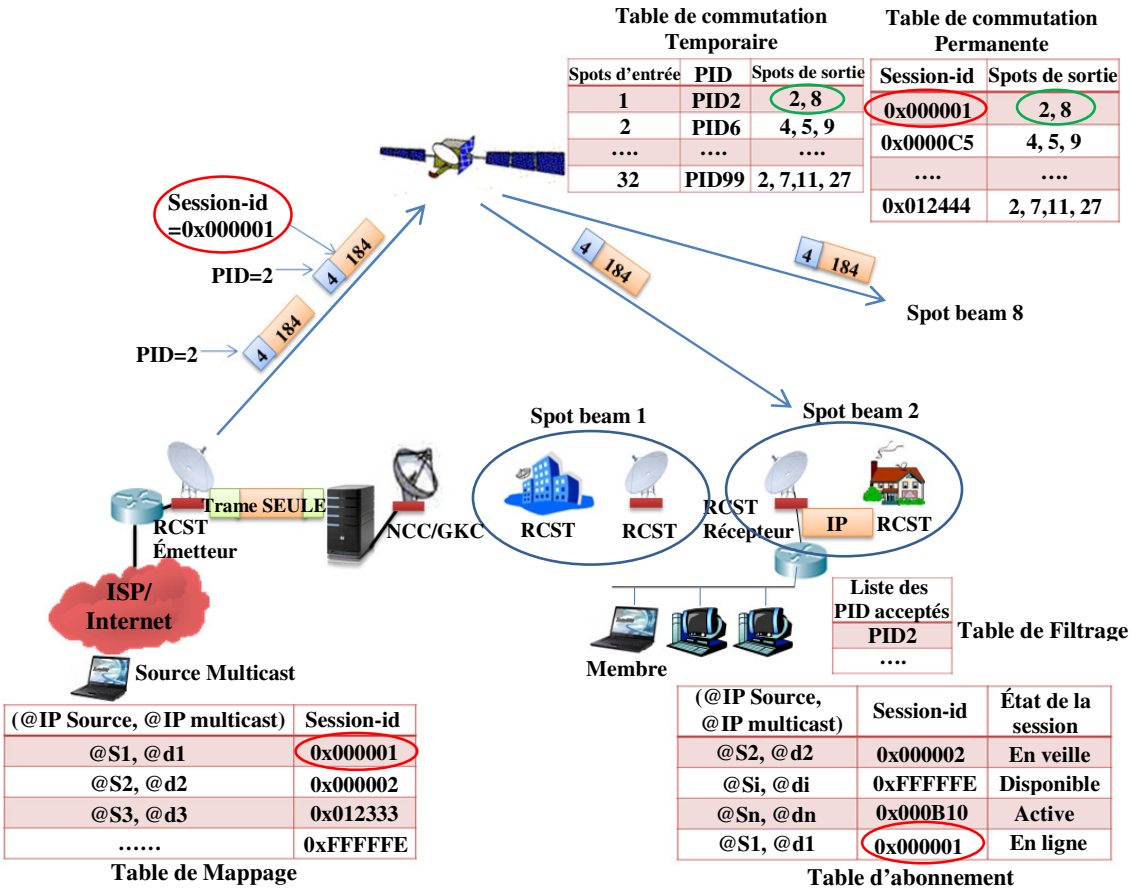

Figure 2.14. Transmission multicast en utilisant l'encapsulation SEULE et l'approche label-switching

Le satellite traite d'abord, l'entête des segments MPEG arrivés pour trouver le PID (par exemple PID=2) et la session-id identifiée par le drapeau PUSI [Clausen *et al.*, 1999] (PUSI=1 seulement du premier segment de n'importe quelle nouvelle trame SEULE). Puis, le satellite teste sa table de commutation permanente pour déterminer la liste des spots de sortie qui conduise aux RCST appartenant à cette session (spot 2 et 8). Ensuite, il met à jour sa table temporaire qui contient les trois champs : le spot d'entrée, le PID, et la liste correspondante des spots de sortie obtenue à partir de la table permanente (spots 2 et 8). Ainsi, les autres segments successifs appartenant à ce paquet multicast seront commutés directement au niveau du satellite selon leur PID, en consultant seulement la table temporaire.

Dans les deux approches (label et self-switching), le RCST récepteur possède une table d'abonnement qui contient toutes les sessions existantes incluant les sessions 'en ligne' correspondantes (c.à.d. les sessions multicast avec des membres existants derrière ce récepteur). Il remplit sa table de filtrage par la liste des PID des sessions 'en ligne' afin de permettre à un segment MPEG reçu d'être décapsulé (accepté) ou non.

Un ensemble des segmentes acceptés seront réassemblés pour former la trame SEULE. Cette dernière sera déchiffrée et authentifiée. Le RCST récepteur extrait alors, le paquet IP original (qui fait partie du trafic multicast) et l'envoie au routeur qui la transmet sur la liaison terrestre.

2.3.4.2 Encapsulation SEULE avec self-switching

En utilisant l'approche 'self-switching', le RCST émetteur doit également utiliser le SEULE pour encapsuler et sécuriser les paquets multicast. Pour cela, il se réfère à sa table de mappage étendue, qui fournit en plus de la correspondance entre les paires des adresses IP et les session-id, le champ *switching-label* pour chaque session active comme montre la figure 2.15. Ainsi, après l'insertion de la session-id et de deux champs dans l'entête EULE, la trame SEULE est segmentée. Ensuite, le RCST émetteur ajoute à l'entête de chaque segment MPEG le *switching-label* correspondant, et envoie ces segments au satellite.

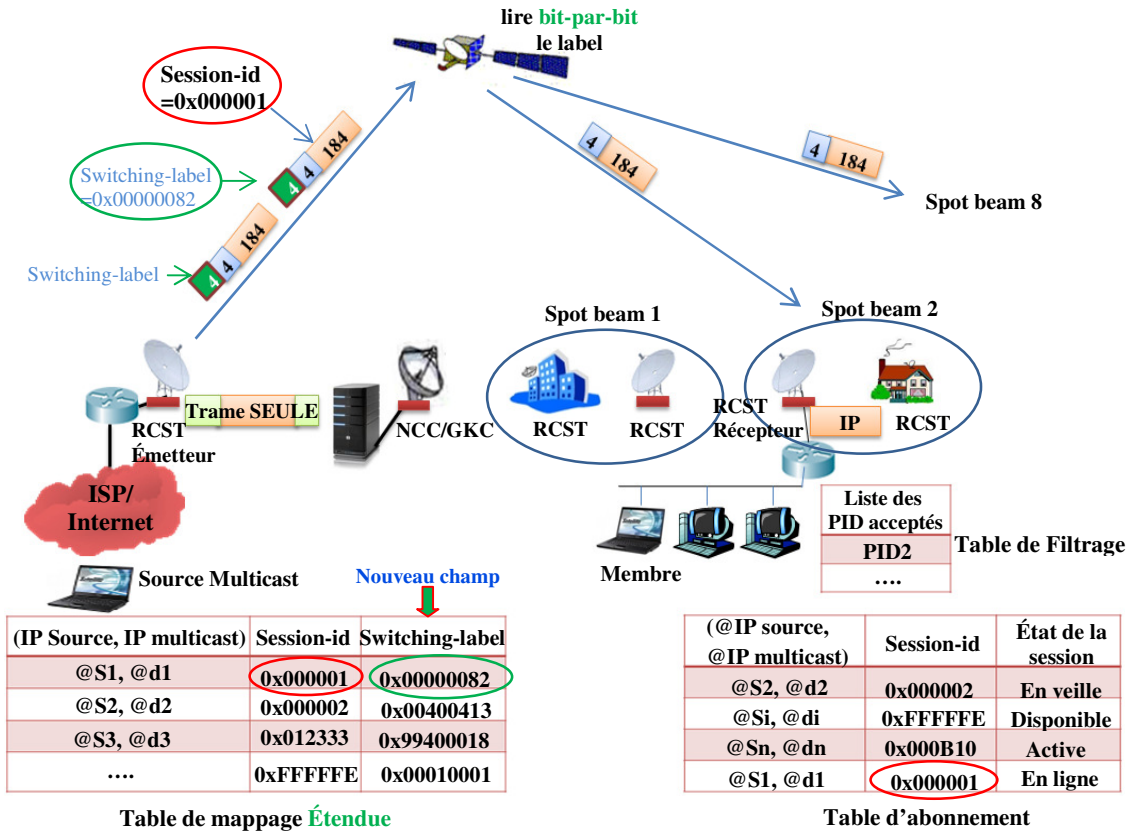

Figure 2.15. Transmission multicast en utilisant l'encapsulation SEULE et l'approche self-switching

Ce denier lit bit-par-bit le *switching-label* de chaque segment arrivé pour trouver les spots de sortie correspondants en testant les bits qui sont mis à 1 dans ce champ. Notons que le satellite enlève le label avant la transmission des segments permettant ainsi l'utilisation des drivers MPEG-2 au niveau des récepteurs et la réduction de la consommation de la bande passante sur la voie descendante du satellite. Les RCST récepteurs se comportent de la même manière que dans le cas de l'approche label-switching.

2.3.5 Système de gestion de clés proposé TLKH (Two-Tiered LKH)

Lord d'un changement dans l'appartenance à un groupe (group membership), la mise à jour des clés s'étend sur tout le système (arbre des clés), et provoque une grande charge sur toutes les ressources du système (GKC, satellite et bande passante de la liaison satellitaire). Par conséquent, de fréquent changement crée une détérioration de performance du système, surtout pour les grands groupes. Plus la dynamicité des membres est élevée, plus la charge et le coût sont grands. Dans l'objectif de réduire ces derniers, nous proposons un nouveau système de gestion des clés à deux couches LKH indépendantes [Ahmad *et al.*, 2012-b] appelé TLKH (Two-Tiered LKH). La première couche se trouve entre le GKC et les RCST (voir figure 2.16). Dans ce cas, les membres GM de l'arbre LKH logique représentent les RCST et non pas les utilisateurs. Aussi, le GKC doit être mis avec le NCC (connecté ou intégré) [Iyengar *et al.*, 2007] au lieu d'être à bord du satellite [Yavuz *et al.*, 2006]. Ceci permet de réduire la charge de génération et de chiffrement des clés au niveau du satellite ainsi que l'espace du stockage des clés.

La deuxième couche se trouve entre chaque RCST et son groupe local des membres (utilisateurs).

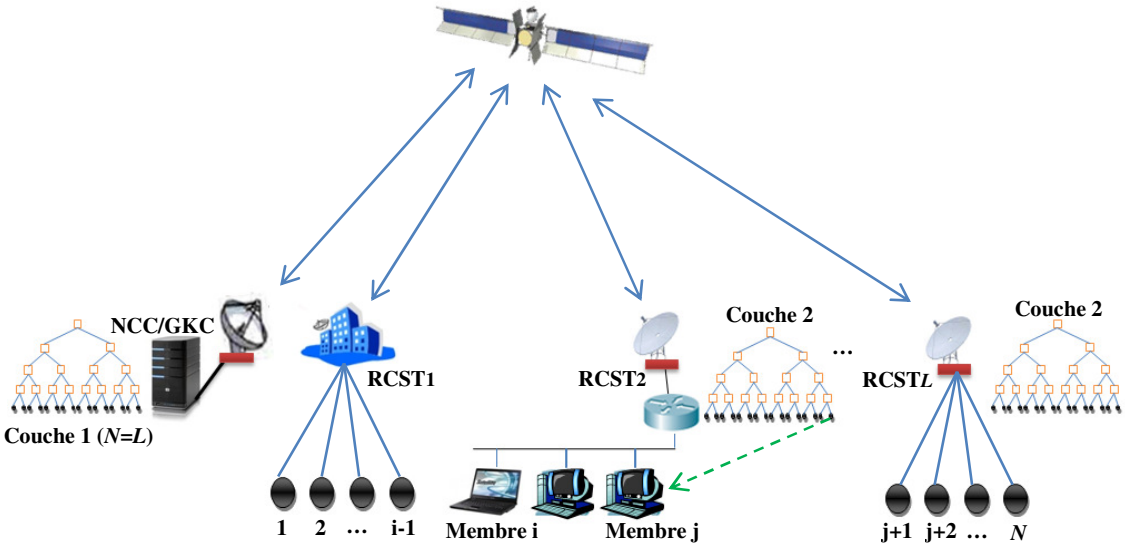

Figure 2.16. Architecture du système avec deux couches LKH indépendantes (TLKH)

En utilisant ces deux couches, un membre qui appartient à un certain RCST (par exemple le membre j du $RCST_2$) et qui souhaite joindre ou quitter un groupe, n'exige que les modifications des clés dans son groupe local terrestre sans affecter les autres RCST ($RCST_1$,.., $RCST_L$) et les transmissions satellitaires correspondantes. Cela réduit énormément la charge sur les ressources du système et offre ainsi un gain de performance significatif surtout sur la liaison satellite.

Les contrôleurs des arbres dans les deux couches, c.à.d. le GKC et chaque RCST, utilise un générateur des séquences chaotiques très performant [El Assad et Noura, 2011], [El Assad *et al.*, 2008], présenté succinctement plus loin, pour la génération de toutes les clés (l'arbre des clés). Ces clés sont chiffrées à l'aide du même algorithme (chaotique ou standard) qui chiffre les trames EULE. Ces générateurs et algorithmes possèdent de très bonnes propriétés cryptographiques.

La taille des clés GK, KEK, et UK est choisie d'une façon à avoir un bon compromis entre la sécurité et le coût de traitement. Nous recommandons une longueur de 128 bits selon les recommandations d'Ecrypt II [EcryptII, 2012] pour les deux clés GK et KEK et une longueur d'au moins 256 bits pour l'UK. Nous proposons également d'utiliser un facteur de branchement (outdegree) optimale *k*=3 dans les deux couches LKH, donnant ainsi le coût minimal de rekeying [Howarth *et al.*, 2004].

2.3.5.1 Système de dérivation des clés transitoires

Un RCST émetteur qui voudrait envoyer le trafic multicast sécurisé, utilise deux clés : TIK (Transient Integrity Key) et TEK (Transient Encryption Key) pour authentifier et chiffrer respectivement chaque trame EULE multicast transmise à travers le satellite. Ces deux clés forment ensemble la clé transitoire TK (Transient Key) qui est dérivée par l'application d'une fonction de hachage (par exemple SHA-256 ou SHA-512) sur : la clé GK, l'identificateur de la session (session-id), et le nonce PN comme indiqué dans la figure 2.17. L'utilisation du PN garantie qu'une nouvelle TK est obtenue pour chaque nouveau paquet (PDU).

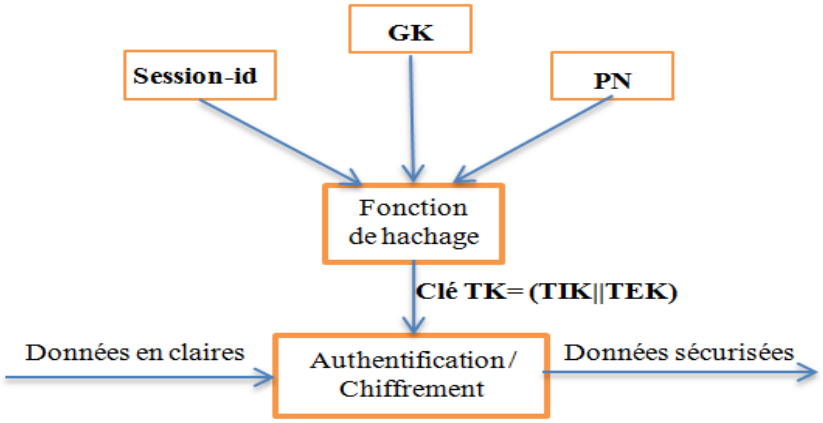

Figure 2.17. Calcul des clés transitoires

D'autre part, la session-id garantie d'avoir deux clés TK différentes pour deux sessions différentes. Un RCST récepteur utilise cette même clé, TK pour le déchiffrement et l'authentification des données. Par ailleurs, l'authentification, le chiffrement ainsi que le hachage peuvent être dynamiques ce qui augmente significativement la sécurité globale des communications.

2.3.5.2 Générateur chaotique proposé pour la génération de clés dynamiques

Nous proposons l'utilisation du générateur chaotique proposé par [El Assad *et al.*, 2008] pour la génération des nouvelles clés de l'arbre {A, B,…, GK}.
Pour obtenir un haut niveau de sécurité, les générateurs chaotiques doivent avoir de bonnes propriétés statistiques dynamiques. Le générateur proposé par [El Assad *et al.*, 2008] est présenté dans la figure 2.18. Il est constitué de deux cellules récursives en parallèles, contenant chacune une fonction non linéaire FNL (x). Différentes fonctions non linéaires FNL(x) ont été testése: $x \exp[\cos(x)]$, $x \cos(x)$, *PWLCM (Piecewise Linear Chaotic Map), Skew Tent map,* etc.

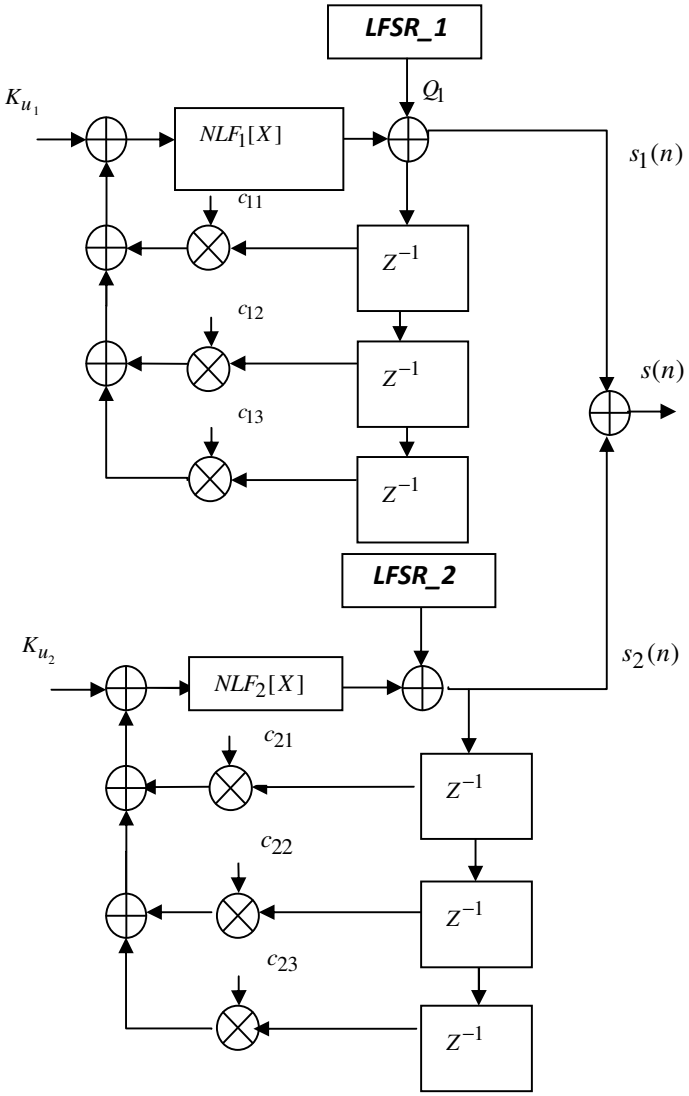

Figure 2.18. Générateur de séquences chaotiques proposé pour la production des clés

Les résultats obtenus par [El Assad *et al.*, 2008] ont montré que les meilleures propriétés cryptographiques des séquences chaotiques générées sont obtenues lorsque les cartes chaotiques discrètes PWLCM et Skew sont utilisées en tant que fonctions non linéaires respectivement par la première et la deuxième cellules récursives. L'équation générale pour ce générateur est définie par les relations suivantes:

$$s_i(n) = NLF_i\{u_i(n-1), p_i\}, i = 1,2 \tag{2.1}$$

$$u_i(n-1) = \mathrm{mod}\left\{\begin{matrix} Ku_i + s_i(n-1)\times c_{i,1} + s_i(n-2)\times c_{i,2} + \\ s_i(n-3)\times c_{i,3}, 2^N \qquad\qquad i=1,2 \end{matrix}\right\}$$

Carte discrète PWLCM:

$$s_1(n) = NLF_1[u_1(n-1), p_1]$$
$$= \begin{cases} \left\lfloor 2^N \times \frac{u_1(n-1)}{p_1} \right\rfloor & if \quad 0 \le u_1(n-1) < p_1 \\ \left\lfloor 2^N \times \frac{2^N - u_1(n-1)}{2^N - p_1} \right\rfloor & if \quad p_1 \le u_1(n-1) < 2^{N-1} \\ NLF_1[2^N - u_1(n-1)] & Otherwise \end{cases} \quad (2.2)$$

Carte discrète Skew:

$$s_2(n) = NLF_2[u_2(n-1), p_2]$$
$$= \begin{cases} \left\lceil 2^N \times \frac{u_2(n-1)}{p_2} \right\rceil & if \quad 0 \le u_2(n-1) \le p_2 \\ \left\lfloor 2^N \times \frac{2^N - u_2(n-1)}{2^N - p_2} \right\rfloor + 1 & if \quad p_2 < u_2(n-1) < 2^N \end{cases} \quad (2.3)$$

Les entrées $k_{u1}(n)$ et $k_{u2}(n)$ sont des paramètres additionnels. Les coefficients c_{ij}, avec $i = 1,2$ et $j = 1,2$ et 3 peuvent prendre des valeurs entre 1 et $2^N - 1$. Les paramètres de contrôle p1 et p2 peuvent prendre des valeurs comprises entre 1 et $2^{N-1} - 1$et 1 et $2^N - 1$ respectivement. Le générateur proposé a été implémenté avec une précision finie *N*=32 bits, alors pour générer une clé de 128 bits, cela nécessite 4 itérations. Les résultats d'évaluation des performances obtenus [El Assad *et al.*, 2008] montrent que le générateur proposé possède de très bonnes propriétés cryptographiques et aléatoires et génèrent des séquences chaotiques performantes.

2.3.5.3 Mécanisme d'identification des membres

Pour garder un bon niveau de sécurité, nous cherchons à transmettre les clés seulement pour le groupe concerné et uniquement aux spots où se trouvent les membres. En LKH, les clés nécessaires pour un processus de rekeying sont transmises un par un, en unicast quand le destinataire est un membre unique et en multicast quand le destinataire est un ensemble de membres. Dans ce dernier cas s'il y a deux clés à envoyer à deux ensembles (sous-groupes) différents dans le même groupe, nous devons effectuer deux transmissions multicast distinctes.

Cette façon de transmettre les clés, chaque clé à part, est parfois très coûteuse. C'est pourquoi nous proposons, lorsque c'est possible, d'envoyer plusieurs clés dans une même transmission (regroupés dans un même paquet). Ce qui réduit la consommation de la bande passante puisque l'overhead émis est réduit, sans oublier la rapidité de la distribution des clés et le gain en temps.

Cependant, en transmettant plusieurs clés dans un même paquet, la question qui se pose est comment un sous-groupe (ou un membre) peut savoir quelle clé lui est destinée. Pour ceci, nous proposons d'utiliser un système de numérotation inspiré du CDMA (Code Division Multiple Access) et du [Harney *et al.*, 2006], qui consiste à attribuer un numéro (série de bits appelé

l'identité ID) pour chaque nœud logique de l'arbre. De cette manière, les membres fils (child members) sont identifiés par l'ID de leur nœud parent.

En conséquence, lorsqu'on transmet plusieurs clés (chiffrées) dans un même paquet, il suffit d'ajouter un indice pour chaque clé qui sert à indiquer le(s) destinataire(s) concerné(s). Cet indice n'est que l'identité ID du sous-groupe ou du membre concerné par la clé. Lorsqu'un membre de la session reçoit un paquet qui ne contient pas des clés qui lui sont destinées, il détruit tout simplement ce paquet. Il est à noter que parfois nous devons effectuer des transmissions unicast pour certains membres. Dans ce cas les clés sont envoyées dans des paquets unicast avec l'adresse NPA@ du destinataire correspondant.

À propos de la numérotation des nœuds, on commence par attribuer le bit 1 pour la racine de l'arbre. Puis, en passant à la couche inférieure, on attribut aux nœuds fils l'ID du nœud parent avec l'ajout de $\lceil log_2 k \rceil$ bits différents, où $\lceil x \rceil$ est le plus petit entier non inférieur à x. La figure 2.19 montre un exemple où k=3 et donc $\lceil log_2 k \rceil$ est égal à deux. En général, la taille (en bits) de l'ID des nœuds des membres est donnée par: $\lceil log_2 k \rceil \times (\lceil log_k N \rceil - 1) + 1$. Lorsqu'un membre (RCST ou un abonné terrestre) se joint à une session, on lui envoie directement son identité ID.

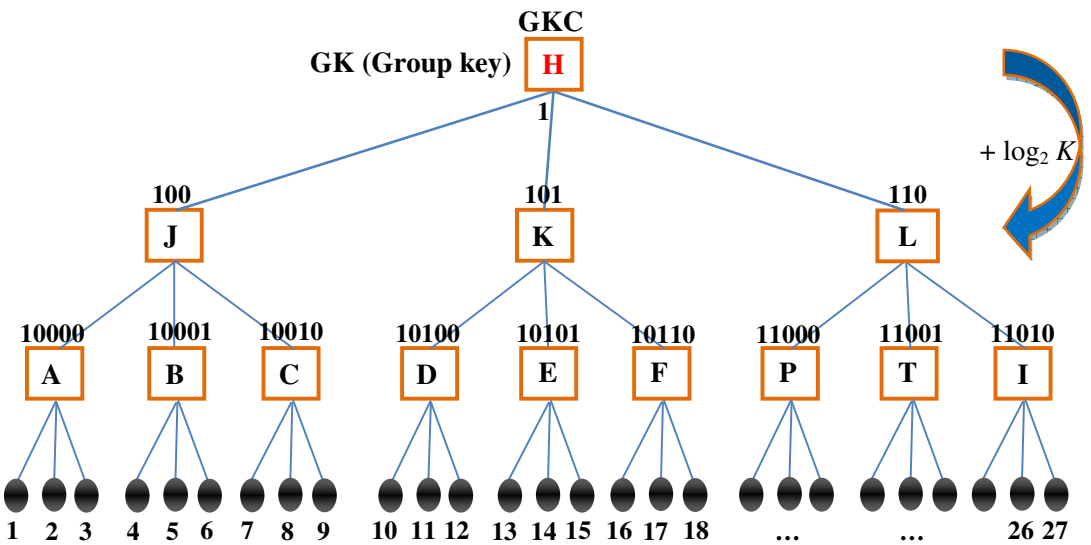

Figure 2.19. Numérotation des nœuds LKH et identification des membres

2.3.5.4 Paquet des clés et des paramètres de sécurité proposé (Key PDU)

Pour permettre la transmission des clés et paramètres de la gestion des clés (clés et paramètres de sécurité), nous proposons un nouveau type de paquet, appelé Key PDU (KPDU), pour le transport des nouvelles clés générées et pour le choix des algorithmes qui seront utilisés pour : la dérivation des clés, le chiffrement et l'authentification [Ahmad *et al.*, 2011]. La structure du KPDU représentée dans la figure 2.20, est similaire à la structure d'un paquet (PDU) normal. Il est constitué d'un entête qui portera les informations relatives à l'association de la sécurité (AS) courant, et d'un corps (payload) qui portera les nouvelles clés. Chacune des clés transportées, KEK*e* et/ou GK*e* est chiffrée avec la clé de sa couche inférieure de l'arbre LKH.

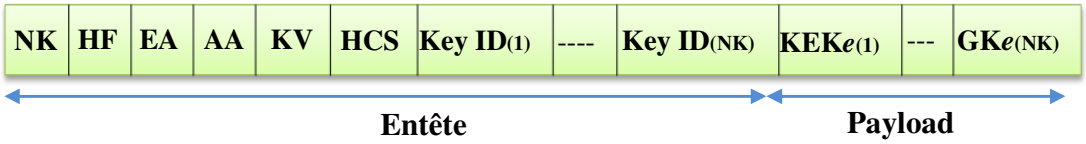

Figure 2.20. Structure de KPDU

L'entête de KPDU contient les champs suivants:

- *NK, Number of keys*: un champ de 4 bits indiquant le nombre de clés transportées par le KPDU: soit le KEK$_{UKi}$ seulement, soit quelques KEK*e*, soit quelques KEK*e* et GK*e*, ou GK*e*.
- *HF, Hash Function*: Un champ de 4 bits indiquant la fonction de hachage utilisée dans la dérivation de la clé transitoire TK.
- *EA, Encryption Algorithm:* un champ de 4 bits spécifiant l'algorithme de chiffrement à utiliser. Cet algorithme sera utilisé pour le chiffrement/déchiffrement de toutes les données envoyées sur la voie satellitaire: les données pour les membres, les données de sécurité (clés), et autres données qui peuvent être portées par les trames EULE.
- *AA, Authentication Algorithm:* un champ de 4 bits indiquant l'algorithme d'authentification à appliquer (par exemple HMAC, OMAC, etc.).
- *KV, Key Version:* un champ de 16 bits qui identifie l'association de sécurité qui est en cours de création. Cette valeur sera utilisée en cas de pertes de la synchronisation de clés entre le GKC et le(s) RCST ou bien entre le RCST et ses membres. Dans le premier cas, le RCST envoie un message d'alarme au GKC indiquant le dernier *Key version* synchronisé. Le format de ce message sera décrit dans le paragraphe suivant.
- *HCS, Header Check Sum:* un champ de 8 bits pour protéger les différents champs de l'entête de KPDU contre la corruption, (similaire à celui de l'entête du paquet IP).
- *Key ID (i)*: un champ sur 16 bits représentant l'ID du nœud parent, qui identifie le(s) membre(s) de réception du ième clé KEK*e*(i) (ou GK*e*(i)) qui se trouve dans le payload. Sa taille peut servir des millions de membres puisque les abonnés (de DVB-S) sont regroupés dans des arbres où chaque arbre terrestre correspond à un RCST. Quand un *Key ID* a une petite taille (moins de 16 bits), nous ajoutons des zéros aux bits les plus signifiants afin d'obtenir un champ de 16 bits.

Nous remarquons que le KPDU offre un mécanisme qui permet une modification dynamique des algorithmes (hachage, chiffrement et authentification). Cela augmente considérablement la sécurité des communications. Enfin le KPDU sera encapsulé et transporté comme n'importe quel autre PDU. Pour rendre cela possible, IANA doit affecter une nouvelle valeur pour le champ *T1* afin d'indiquer ce nouveau type de paquet.

2.3.5.5 Message d'alarme DULM

Il peut y arriver que le RCST perde la synchronisation des clés avec le NCC/GKC. La perte peut être due à différentes circonstances telles que le niveau élevée de bruit sur le canal de transmission, une panne matérielle, une panne de courant, attaques actives ou autres raisons. Dans ce cas, le RCST récepteur est incapable de récupérer les nouvelles clés ou de déchiffrer n'importe quel paquet. Il informera le GKC sur le problème en lui envoyant un message d'alarme à travers le canal de retour (DVB-RCS) afin de resynchroniser les clés.

Pour traiter ce message de signalisation provenant du RCST récepteur, nous proposons d'utiliser le message DULM (Data Unit Labelling Method). Ce dernier est un message de base qui permet aux RCST d'envoyer des informations de contrôle et/ou de gestion au NCC/GKC. Il peut contenir différents types d'éléments d'informations (IE, Information Element) avec différentes structures indiquées par le champ "IE TYPE" (voir figure 2.21). Ce champ offre la possibilité de définir de nouveaux genres de messages en utilisant les valeurs de ses types réservés [ETSI 301 790, 2009]. Pour cela, nous proposons d'utiliser un nouveau "IE TYPE", avec un id égal à 0x19 (valeur réservée) pour définir notre message d'alarme. Son format spécifique, donné par la figure 2.21, est conçu pour assurer la resynchronisation des clés.

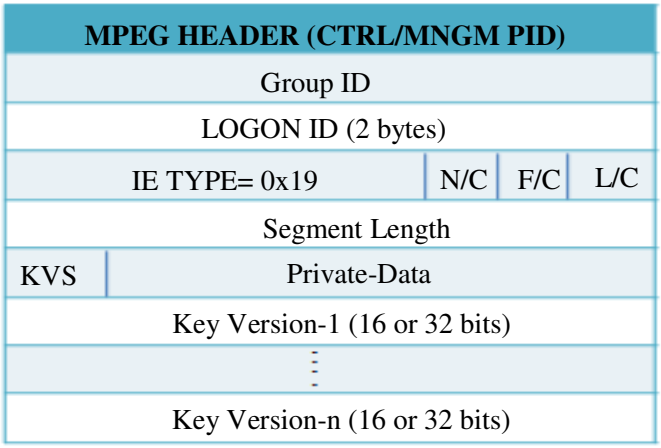

Figure 2.21. Format du message d'alarme DULM

Ce message est envoyé en utilisant un PID égal à CTRL/MNGM PID, qui est permis à être utilisé par chaque RCST dans la phase de connexion. Dans le nouveau message d'alarme proposé, nous suggérons d'ajouter trois nouveaux champs à ceux déjà définis dans le standard DVB-RCS [ETSI 301 790, 2009]. Le premier est un champ d'un seul bit, appelé *KVS* (Key Version Size). Il définit la taille du champ *Key Version*, et il est égal à 0 pour une taille de 16 bits et 1 pour 32 bits (ce dernier est utilisé pour les systèmes ayant un taux élevé d'opérations de joindre/quitter).

Le deuxième champ est le *Private-Data* qui est un champ réservé à l'usage de l'opérateur. Le dernier champ est le *Key Version-n* qui porte le numéro du dernier KPDU valide reçu pour la session en ligne *n*. C'est le champ qui permet au GKC de connaître le point de rupture et de rétablir la synchronisation en envoyant les messages KPDU appropriés.

2.3.6 Modification de la structure de l'encapsulateur/décapsulateur ULE

Pour implémenter notre système de sécurité, il faut faire une petite modification sur la structure de l'encapsulateur du RCST émetteur et sur la structure de décapsulateur ULE du RCST récepteur. En effet, la méthode EULE proposée permet l'utilisation des encapsulateurs ULE existants en procédant à une légère modification pour tenir compte des nouveaux champs proposés. Donc, nous n'avons pas besoin de réaliser de nouveaux encapsulateurs, mais seulement une mise à jour sur l'encapsulateur IP standard pour tenir compte des exigences de notre méthode EULE. Cette modification consiste à ajouter la *Table de mappage (Mapping Table)* à la structure de l'encapsulateur IP formé par: *le Tampon PDU (PDU Buffer),* le *Créateur SNDU (SNDU Creator)* et *l'emballage MPEG (MPEG Packager)* [Caragata *et al.*, 2010]. La figure 2.22 représente l'encapsulateur IP modifié.

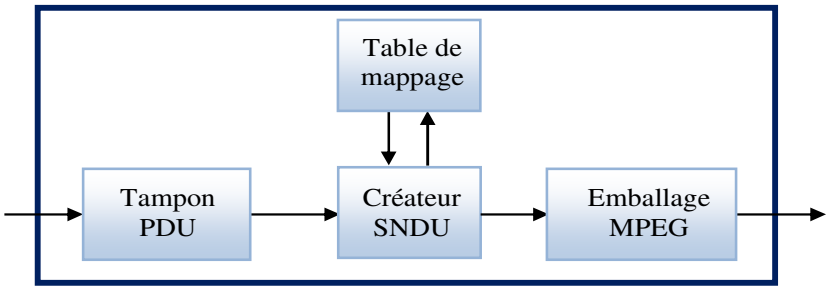

Figure 2.22. Structure de l'encapsulateur IP modifié

La fonction de chaque bloc est:

- *Tampon PDU* : mémorise les PDU qui attendent d'être encapsulés et transmis.

- *Table de mappage* : fournit la valeur de la *session-id*, réponse à une requête demandée par le créateur SNDU et contenant l'adresse IP source et l'adresse IP d'un groupe multicast (destinataire).

- *Créateur SNDU* ou *EULE* : ajoute au PDU l'en-tête et l'en-queue nécessaires pour former la trame SNDU (ou la trame EULE), qui sera ensuite emballée dans des segments MPEG-TS avant d'être envoyée. Ce bloc détermine les champs suivants (voir figures 2.4 et 2.12): la longueur du SNDU, le type de la PDU, l'adresse de destination unicast (si nécessaire) ou la *session-id* avec le *FEC* en cas de multicast, et enfin le code de contrôle de redondance cyclique (CRC).

- *Emballage MPEG* : arrange la SNDU dans des segments MPEG-TS. Le créateur SNDU et l'emballage MPEG doivent respecter la norme [Fairhurst et Collini-Nocker, 2005].

Pour sécuriser les trames EULE et introduire la composante de la gestion des clés offerte par notre système de sécurité proposée, la structure de l'encapsulateur modifié, située au niveau du RCST émetteur, est décrite dans la figure 2.23. Dans cette nouvelle structure [Ahmad *et al.*, 2012-c], le *tampon PDU*, la *table de mappage*, et *l'emballage MPEG* restent inchangés. Le *Créateur SNDU* est légèrement modifié puisqu'il a une tâche supplémentaire qui est d'ajouter un entête

d'extension à l'EULE. Il est connecté au bloc *Créateur des clés* (*Key Creator*) pour fournir le PN et la session-id. Les nouveaux blocs ajoutés sont les suivants :

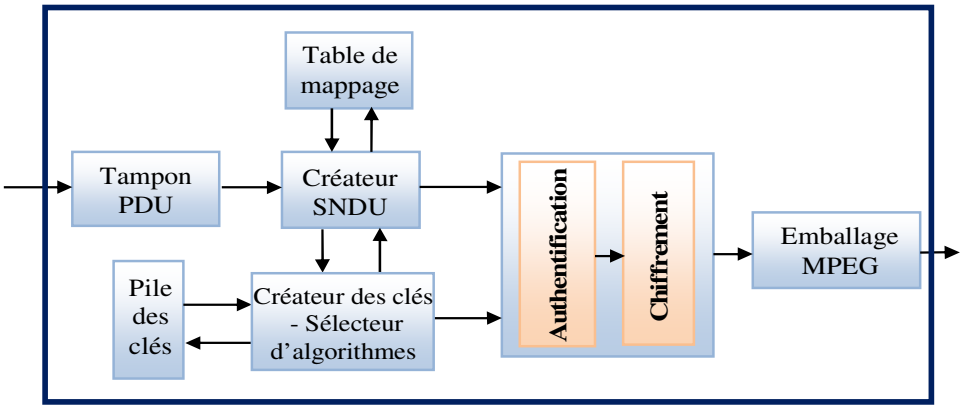

Figure 2.23. Encapsulateur EULE sécurisé au niveau du RCST émetteur

- *Pile des clés (Key Stack)*: il est en contact avec le GKC en vue d'obtenir les clés (KEK et GK) de l'arbre LKH avec les algorithmes spécifiés. Son rôle est de fournir au bloc suivant les types d'algorithmes et la clé du groupe multicast (GK).
- *Créateur des clés et Sélecteur d'algorithmes (Key Creator - Algorithm Selector)*: Ce bloc a pour rôle de générer la clé transitoire (TK) et de sélectionner les algorithmes de: hachage, chiffrement, et d'authentification.A cet effet, les blocs *Créateur SNDU* et *Pile des clés* lui offrent les champs nécessaires.
- *Authentification/Chiffrement (Authentication/Encryption)*: effectue l'authentification en déterminant le code MAC qui couvre l'entête du SEULE et le PDU. Ensuite, il chiffre ce dernier ainsi que le MAC obtenu avec l'algorithme de chiffrement sélectionné. Les clés secrètes ainsi que les types d'algorithme d'authentification et de chiffrement à utiliser sont précisés par le bloc *Pile des clés*.

La structure du décapsulateur (désencapsuleur) qui se trouve dans les RCST récepteurs est montrée dans la figure 2.24. Les blocs *Pile des clés* et *Créateur des clés* sont similaires à ceux existants dans l'encapsulateur. Le bloc *récupérateur SNDU* (*SNDU Recovery*) extrait les trames EULE sécurisées (SEULE) des segments MPEG, et il les envoie au bloc Déchiffrement/ Authentification pour être déchiffrées et authentifiées.

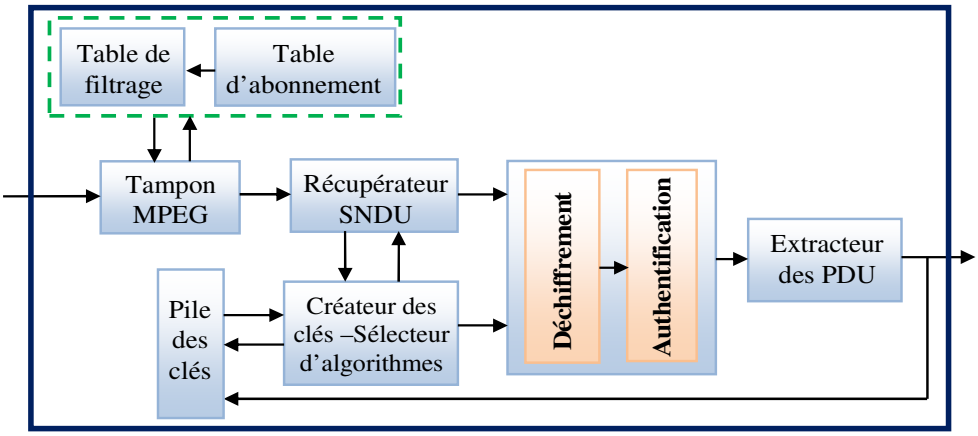

Figure 2.24. Décapsulateur EULE sécurisé au niveau du RCST récepteur

Le *tampon MPEG* (*MPEG Buffer*) rejette ou laisse passer les segments MPEG-2, en se basant sur les PID obtenus à partir de la *table de filtrage* (Filtering Table), et sur la *Table d'abonnement* qui identifie les sessions acceptées. *L'extracteur des PDU* extrait le PDU ou le KPDU du SNDU (ou EULE) reçu. Il opère comme suit : Les PDU sont envoyés au routeur qui transmet les paquets multicast sur la liaison terrestre, et les KPDU sont envoyés vers le bloc *Pile des clés* qui déchiffre les nouvelles clés requises avec l'algorithme spécifié et il les stocke.

La structure sécurisée de l'encapsulateur/décapsulateur proposée intègre, en plus par rapport au système standard, les fonctions de mappage et de filtrage, de création et de gestion des clés, d'authentification et de chiffrement.

2.4 Analyse des performances du système proposé et résultats de simulation

2.4.1 Analyse et avantages du système de gestion des clés proposé

Puisque les ressources satellitaires sont coûteuses, il est important de réduire la consommation de la bande passante et la charge de travail des composantes du système. Ces dernières seront réduites significativement par notre système de gestion des clés proposé. En effet, l'analyse comparative des performances faite ci-dessous entre le système de gestion des clés proposé TLKH et les systèmes Flat et LKH, confirme nos propos.

La comparaison est faite en utilisant les deux critères suivants : le premier critère qui est le plus important représente le coût de rekeying (renouvellement des clés) sur la liaison satellitaire qui détermine aussi l'effort de calcul et la charge de travail des composantes du système ; le deuxième critère représente le nombre des clés stockées dans le GKC et dans les autres composantes : les RCST et les membres.

2.4.1.1 Coût de rekeying (renouvellement des clés)

Définissions d'abord le paramètre de la volatilité (ou dynamicité) α, comme étant le nombre moyen de rekeying par un membre GM. Donc, une valeur de α=1 indique qu'il y a, en moyenne, une seule opération de rekeying par chaque GM pendant la durée de vie du groupe sécurisée. Donc avec N membres, le nombre de rekeying est donné par $N \times \alpha$.

Les coûts de rekeying des protocoles Flat et LKH sont donnés par N et $klog_k N$ respectivement [Howarth *et al.*, 2004]. Puisque les membres sont gérés seulement par un contrôleur de groupe centralisé (GKC) dans les deux protocoles, le lien satellitaire et les composantes du système seront affectés par chaque opération de rekeying (renouvellement des clés). Ainsi, pour $N \times \alpha$ rekeying, le coût total de rekeying à travers le satellite devient $N^2 \times \alpha$ et $(k\ log_k N) \times N \times \alpha$ pour les protocoles Flat et LKH respectivement.

Avec l'architecture proposée de TLKH, le GKC n'est responsable que du renouvellement des clés (rekeying) pour les RCST et non pas pour les membres (utilisateurs) directement. Ainsi, les ressources satellitaires sont presque non affectées par le rekeying provoqué par les membres. Le rekeying d'un RCST se fait lorsqu'il reçoit le premier membre ou lorsque tous ses membres quittent le groupe. Et dans ce cas seulement, la liaison satellitaire est affectée par le rekeying. Cette situation réduit considérablement le trafic de la gestion des clés à travers la liaison satellitaire et la charge de travail des composantes du système. Donc, contrairement aux membres, les RCST ne montrent pas un caractère dynamique de 'joindre/quitter' et la charge de rekeying provenant des RCST est pratiquement négligeable.

Ayant L RCST, le coût de rekeying à travers le satellite dans notre système est $(k\ log_k L) \times m \times L$, où m représente la dynamicité des RCST (similaire au paramètre α). Il est clair que m est nettement plus petit que α, dans l'ordre de $\alpha/10$, $\alpha/50$ ou $\alpha/100$.

La figure 2.25 montre le coût de rekeying à travers la liaison satellitaire en fonction de α pour les trois protocoles, Flat, LKH et TLKH. Pour la simulation, nous avons pris les valeurs N=10^6 (pour un très grand système multicast), k=3 et L=500. Pour TLKH, nous avons sélectionné trois valeurs pour m : $\alpha/10$, $\alpha/50$ et $\alpha/100$.

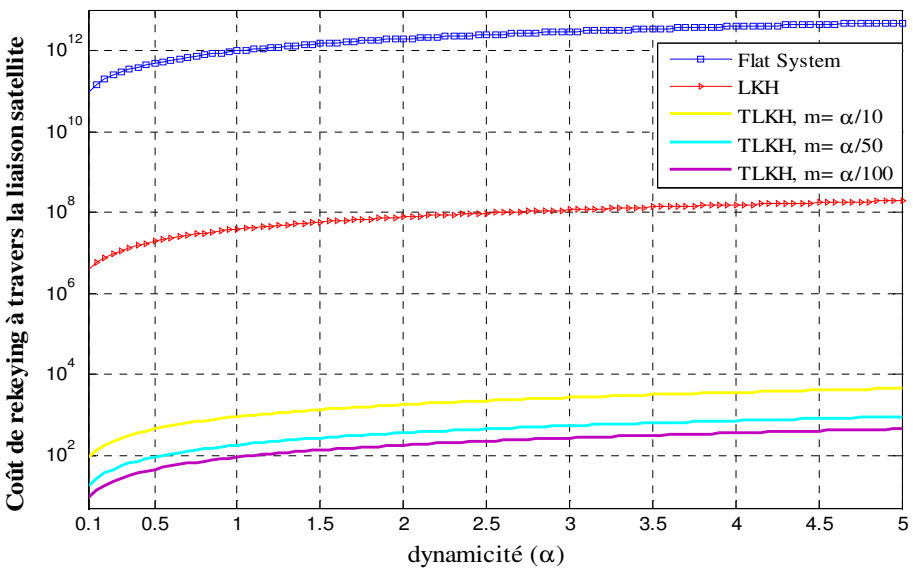

Figure 2.25. Coût de rekeying à travers la liaison satellitaire dans les différents systèmes

Lorsque α augmente, le coût dans les trois protocoles augmente, mais il est clair qu'il y a une grande différence entre ces protocoles. Par exemple, pour α=0.5, le coût du protocole Flat est d'environ 5×10^{11} transmissions, celui de LKH est d'environ 2×10^{7} et le coût de notre système varie approximativement entre 45 (pour $m= \alpha/100$) et 450 (pour $m= \alpha/10$). Il est clair donc qu'il y a un gain significatif de bande passante sur la liaison satellite car le trafic de gestion des clés est significativement réduit. Ceci montre que, l'impact de la dynamicité des membres sur la liaison satellitaire est presque négligeable dans notre système.

Dans la couche terrestre, le coût de rekeying au niveau d'un RCST peut être trouvé de la même manière. Chaque RCST gère un group local des membres comportant en moyenne $n_l \cong N/L$ membres. Chaque fois qu'un événement d'un 'membre qui se joindre/quitter' survient, le RCST applique la règle LKH sur son groupe local. Ainsi, le nombre de rekeying pour un groupe d'un RCST est $r_l = N\times\alpha /L$, et le coût de rekeying au niveau d'un RCST est $(klog_k\, n_l) \times r_l$. Pour α=0.5 et pour les mêmes valeurs considérées dans la figure 2.25, le coût de rekeying d'un RCST est approximativement 21000. Ce coût est facilement manipulable même par un RCST à faible capacité.

2.4.1.2 Nombre de clés stockées dans les différentes composantes

Le nombre de clés stockées dans le GKC est le deuxième critère important à étudier. Dans le protocole Flat, le GKC doit stocker une clé unique pour chacun des membres avec la GK, ainsi la charge de stockage est N+1. Dans le protocole LKH, le GKC stocke (kN-1)/(k-1) clés [Ng et Sun, 2005]. Dans notre système de gestion des clés, le GKC stocke les KEK, une clé pour chaque RCST, et la GK. Ainsi, le nombre de clés stockées dans le GKC est (kL-1)/(k-1)$\approx 3L/2$ pour k=3, ce qui est clairement plus petit que les protocoles Flat et LKH.

Pour les mêmes valeurs de N, k et L mentionnées dans le paragraphe précédent, le nombre de clés qui sont stockées dans le GKC est 10^6, 1.5×10^6 et 750 pour les protocoles Flat, LKH et TLKH respectivement. Ceci montre la grande différence entre notre système et les deux autres protocoles.

Le nombre de clés qui sont stockées dans un RCST, est composé du vecteur des clés LKH de la couche satellitaire conjointement avec les clés LKH des ses membres dans la couche terrestre: (kn_l-1)/(k-1) $+log_k L$. Le nombre de clés qui sont conservées dans chaque membre en Flat et LKH est 2 et $log_k N$ +1 respectivement, alors qu'il est égal à $log_k n_l$ +1 dans TLKH.

2.4.1.3 Résumé des avantages de TLKH par rapport aux systèmes Flat et LKH

Le tableau 2.1 suivant résume les avantages du protocole TLKH par rapport aux protocoles Flat et LKH en termes de coût de rekeying à travers la liaison satellitaire et en termes du nombre de clés stockées dans le GKC et dans les membres.

Avantages de TLKH par rapport au Flat et LKH	**k=3, L≥500, α ≥0.1 et N≥10^5, $n_l = N/L$**				
	Coût de rekeying à travers la liaison satellite	Coût de rekeying au niveau d'un RCST	Nombre de clés stockées dans GKC	Nombre de clés stockées dans le RCST	Nombre de clés stockées dans un membre
Système Flat	$N^2 \times \alpha$	-	N+1	-	2
LKH	$(k\ log_k N) \times N \times \alpha$	-	$\frac{(kN-1)}{(k-1)}$	-	$log_k N$ +1
TLKH	$(k\ log_k L) \times m \times L$,	$(k log_k n_l) \times r_l$, $r_l = N \times \alpha / L$	$\frac{(kL-1)}{(k-1)} \approx \frac{3L}{2}$	$\frac{(kn_l-1)}{(k-1)} + \log_k L$	$log_k n_l$ +1

Tableau 2.1. Comparaison des performances du système TLKH avec les systèmes Flat et LKH

2.4.2 Analyse de la consommation de la bande passante

Dans ce paragraphe nous analysons les performances du système proposé, selon le critère de la consommation de la bande passante des données de la gestion des clés et du taux des données ajoutées par les services de sécurité et de commutation. Cette analyse permet, d'un côté de prouver que les caractéristiques du système proposé sont très bonnes et, d'autre côté, de fournir toute l'information nécessaire à un fournisseur de services IP par satellite afin de lui permettre de choisir l'approche adéquate.

Le trafic des données de la gestion des clés et le taux des données ajoutées sont des paramètres importants pour un système de communications satellitaire, puisque la ressource spectrale est rare et chère, et elle doit être utilisée de manière très efficace. En effet, un nombre limité des transpondeurs, qui ont une largeur de bande limitée approximativement à 40 MHZ, sont disponibles. Donc pour que le système de communication soit fiable, la ressource spectrale doit être utilisée de manière optimale.

2.4.2.1 Données de la gestion des clés

Pour calculer le trafic des données de la gestion des clés transmis dans un processus de rekeying, nous devons connaître le nombre de messages à transmettre ainsi que le nombre de clés transportées dans ces messages. Dans notre système, les nouvelles clés nécessaires pour un rekeying sont partagées dans un ensemble des paquets KPDU transmis sur la liaison satellitaire ou terrestre. Soit *NT* le nombre de ces paquets transmis. Il est égal à $\lceil Nt \rceil$, avec *Nt* donné par :

$$Nt = \begin{cases} k + 0.5 \lceil \log_k N \rceil - 1.5 & \text{(Quitter)} \\ k + 0.5 (\lceil \log_k N \rceil - 1) & \text{(Joindre)} \end{cases} \qquad (2.4)$$

Pour voir comment les clés sont distribuées dans les paquets transmis, prenons d'abord le cas où un membre quitte son groupe. Si *Nt* est un entier (Є N), les clés seront répartis dans les paquets comme suit : nous avons (*k*-1) paquets, contenant chacun une seule clé, et $(\lceil \log_k N \rceil - 1)/2$ paquets avec 2*k* clés chacun. Si *Nt* est un nombre décimal (Є Q), les clés seront distribuées de la manière suivante : (*k*-1) paquets avec une seule clé, $(\lceil \log_k N \rceil - 2)/2$ paquets contenant chacun 2*k* clés et un paquet avec *k* clés.

Dans le cas où un nouveau membre joint le groupe, un paquet supplémentaire contenant la clé appropriée doit être envoyé à ce nouveau membre. Et ainsi, *k* paquets sont transmis avec une clé au lieu de (*k*-1) paquets dans le cas précédent.

A titre d'exemple nous prenons l'arbre de la figure 2.11 et nous supposons que le membre GM1 veut quitter son groupe. Le nombre des paquets KPDU nécessaires pour faire le rekeying est donné par *Nt* (Quitte), qui est égale à 2.5 dans ce cas. Ainsi, *NT*=3 paquets qui sont nécessaires et *Nt* est un nombre décimal. Donc, la distribution des clés sur ces paquets sera comme suit : le premier paquet transmis contient seulement le $\{F_{UK12}\}$, le deuxième paquet contient les 4 clés $\{K_E, K_F, N_K, N_L\}$ et le dernier paquet contient les 2 clés $\{O_M, O_N\}$. Dans le protocole LKH classique où un seul arbre est utilisé pour tout le DVB-S, nous avons besoin d'envoyer un message pour chaque clé. Pour l'exemple considéré, puisque nous avons à transmettre 7 clés, le système LKH classique doit effectuer 7 transmissions contre 3 transmissions seulement pour notre système. En général, le système proposé réduit le nombre des messages de rekeying transmis de $klog_k N$-1 à *NT* en cas de 'départ' d'un membre et de $klog_k N$ à *NT* en cas de 'se joint'. La figure 2.26 montre la variation dans le nombre de transmissions des messages de rekeying entre le LKH classique et le système proposé, en fonction du nombre total des membres *N* (avec *k*=3 et *L*=500). Ces formules et graphes sont valables pour les deux couches, satellite et terrestre.

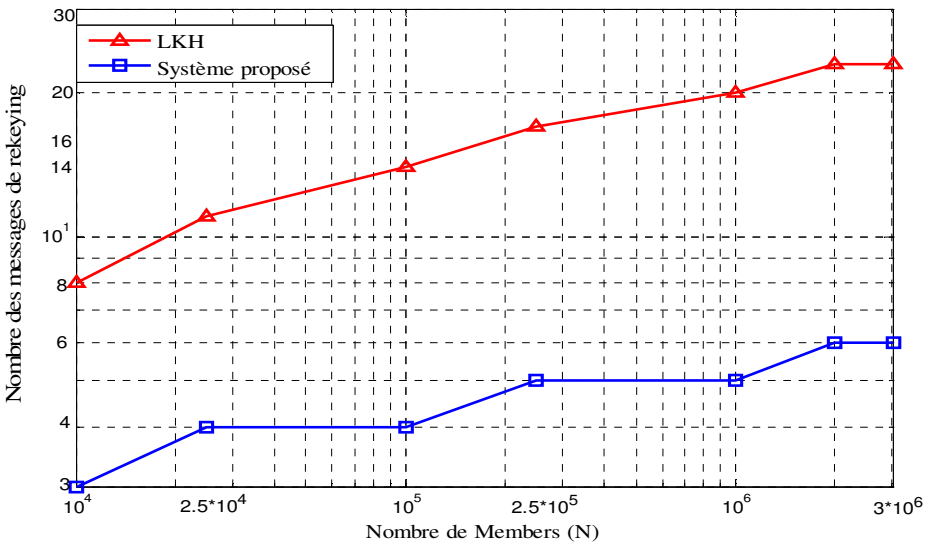

Figure 2.26. Nombre de transmissions pour un rekeying

La figure ci-dessus montre que le nombre des messages de rekeying pour notre système est distinctement plus petit que celui utilisé par LKH. Par exemple, pour N=10^6 membres (avec n_l =2000 membres appartenant à chaque RCST), le nombre des messages de rekeying dans la couche terrestre est égal à: 20 transmissions pour le LKH classique et 5 transmissions pour notre système. Dans la couche satellitaire, où L=500 RCST, le nombre de transmissions est : 17 pour LKH et 5 pour notre système. Alors, la réduction dans le nombre des messages de rekeying est en moyenne égale approximativement à 72%.

Même si on diminue le nombre des messages de rekeying, ceci n'aura pas trop d'importance si l'overhead de ces messages augmente. Donc, pour démontrer la supériorité du système proposé, nous devons analyser le trafic des données de la gestion des clés transmis *KMD* (Key Management Data), qui représente la consommation de la bande passante. Pour un rekeying, le trafic total des données de la gestion des clés *KMD* transmis à travers l'une des deux couches, satellite ou terrestre, est donné (en octets) par :

$$KMD = [(TNK) \times (KL + KIS) + (NT) \times (FKH + EH) \times 8]/8 \qquad (2.5)$$

Où:

- *TNK, Total Number of Keys*: est le nombre des clés transmises (qui est égale aux nombre de transmissions effectués par le LKH classique)
- *KL, Key Length*: est la longueur de chacune des clés transmises (nous recommandons une longueur de 128 bits pour chacune des clés KEK et GK).
- *KIS, Key ID Size*: est la taille de *Key ID* en bits (16 bits).
- *FKH, Fixed KPDU Header length*: est la longueur de la portion fixe de l'entête KPDU.
- *EH, Extension Header*: est la longueur de l'entête d'extension (6 octets).

Nous avons effectués une comparaison entre notre système et la méthode ECPVSS (Elleptic Curve Pintsov-Vanstone Signature Scheme). Cette dernière fournit d'excellente performance [Pintsov et Vanstone, 2000] [Naccache et Stern, 2000] puisqu'elle consomme le minimum de la bande passante par rapport à la meilleure méthode compétitive. Elle offre une taille moyenne minimale d'environ 50 octets pour un message de rekeying. Nous avons simulé le *KMD*, pour notre système et pour l'ECPVSS, en fonction du nombre des membres *N*. Les résultats obtenus sont présentés dans la figure 2.27. Nous remarquons que la différence entre les deux courbes est très nette et elle est presque constante. Le *KMD* dans l'ECPVSS dépasse le double de celui de notre méthode. En utilisant k=3 et L=500, le *KMD* pour N=10^6 par exemple, est 1000 octets pour l'ECPVSS et à peu près 410 octets pour notre méthode.

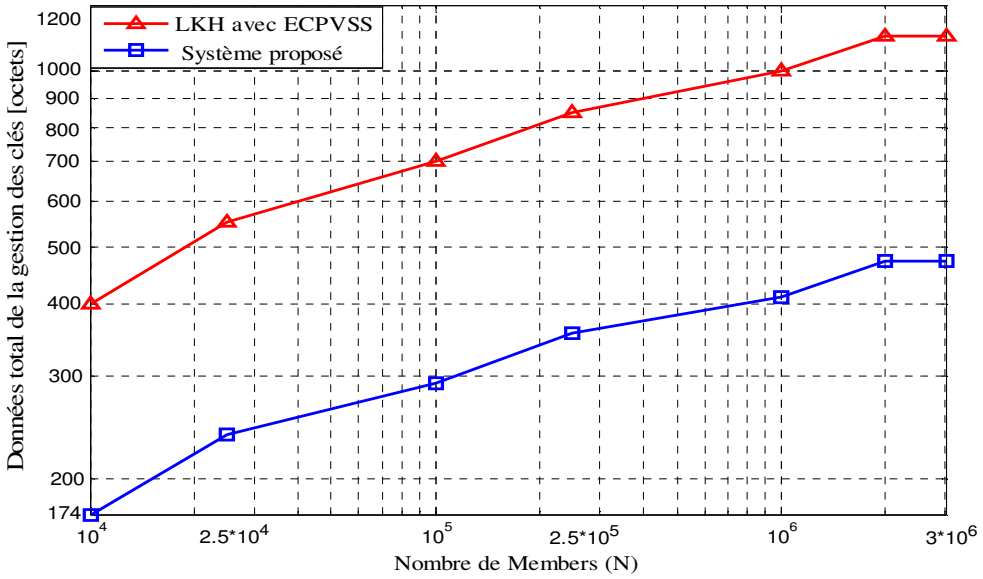

Figure 2.27. Trafic total des données transmises pour un rekeying

Ceci montre que le système proposé offre une réduction approximative de 58% sur l'overhead ajouté ainsi que sur la consommation de la bande passante. Ceci peut être expliqué par le fait qu'un seul overhead est ajouté au paquet KPDU qui transmet un ensemble des clés contre un overhead pour chaque clé transmise dans un paquet ECPVSS.

Ce que nous venons de présenter est pour le cas où un membre quitte son groupe. Pour l'autre cas où un membre se joint à un groupe, les résultats obtenus de la figure 2.27 et l'analyse de performance effectuée ne changent pas trop, car dans ce cas, un seul message supplémentaire est envoyé par rapport au cas précédent, et ce message ne contient qu'une clé unique (de taille 128 bits).

2.4.2.2 Taux des données ajoutées par les services de sécurité et de commutation

Pour notre système multicast considéré, nous définissons le taux des données ajoutées *DO* (Data Overhead) comme la quantité des données ajoutées nécessaires d'être envoyées en utilisant la liaison satellitaire afin d'assurer les services de la sécurité et de du transport efficace du multicast. Le taux *DO* est exprimé comme étant le rapport entre les entêtes ajoutés et la longueur de la trame SEULE. Le *DO* dans notre système comporte deux composantes : la composante de la sécurité et la composante de la commutation. L'overhead provenant de la première composante, appelée *DOse*, est composée de 6 octets ajoutés à l'entête de chaque trame EULE et représentée par les deux champs *T2* et *PN* (voir figure 2.13). Le deuxième overhead provient de la composante de la commutation *DOsw*, elle représente l'information ajoutée par les deux couches encapsulation et MPEG. Ce *DOsw* vari selon l'approche sélectionnée : label-switching ou self-switching.

Pour le label-switching, il n'y a pas de bits supplémentaires ajoutés par la méthode d'encapsulation proposée. Également, la couche MPEG n'ajoute aucun bit. Donc, pour cette approche, l'overhead total *DO* est égal à *DOse*. Il est donné en pourcentage par :

$$DO\ (label\text{-}s) = 100.\ EH\ /\ l(SEULE) \qquad (2.6)$$

Où *EH* est la longueur de l'entête d'extension (6 octets) et *l*(*SEULE*) est la longueur moyenne de la trame SEULE. Puisque la trame SEULE encapsule un paquet PDU (avec un nombre fixe de 20 octets), alors le *DO (label-s)* est une fonction de la longueur moyenne du paquet.

Pour le self-switching, nous ajoutons un champ appelé *switching-label* de taille *SLS* (Switching-Label Size) égale à 4 octets ou plus sur chaque segment MPEG (au niveau du couche MPEG). Comme la taille de chaque segment MPEG est fixée à 184 octets, le nombre de segments nécessaires pour transporter une trame SEULE est égale à $\lceil l(SEULE)/184 \rceil$, où $\lceil x \rceil$ est le plus petit entier non inférieur à x. Ainsi, le pourcentage total de *DO (self-s)* est donné par:

$$DO\ (self\text{-}s) = 100.\ [EH + (\left\lceil \frac{l(SEULE)}{184} \right\rceil) \times SLS]\ /l(SEULE) \qquad (2.7)$$

Le *DO (self-s)* dépend de la longueur moyenne du paquet et du *SLS* ajoutée.

Nous avons simulé le *DO* total pour chaque approche et pour des longueurs moyennes des paquets variant entre 25 et 1500 octets. Les résultats obtenus sont présentés dans la figure 2.28. Pour la simulation de l'approche self-switching, et puisque la taille du label ajoutée dépend directement du nombre des spots-beams utilisé par le satellite, nous avons considéré les deux cas les plus prévalent : un satellite de 32 spot-beams (comme pour "EuroSkyWay") et un satellite de 82 spot-beams (comme pour "Eutelsat's").

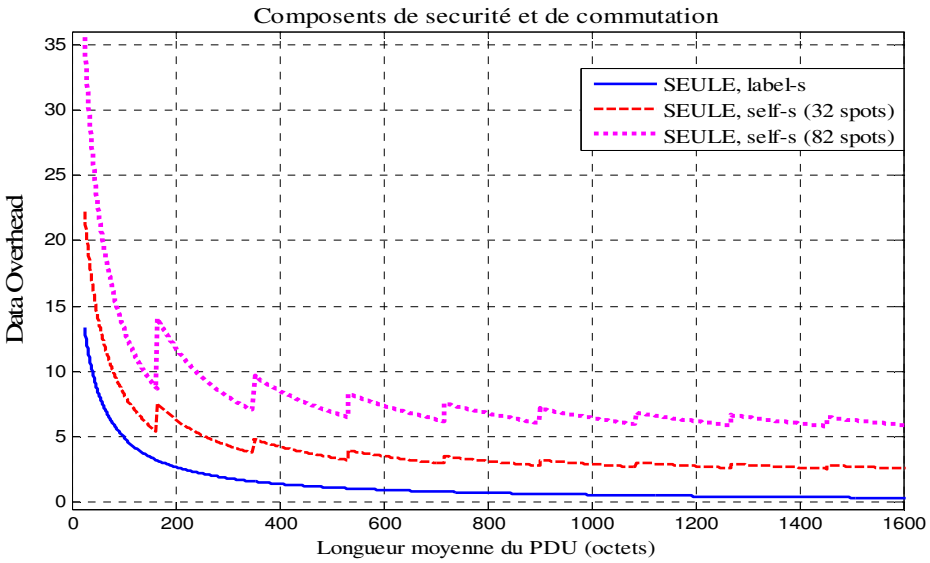

Figure 2.28. Taux total des données ajoutées (DO) pour des longueurs des paquets IP entre 25 et 1500 octets

À partir de la figure 2.28, pour les grandes valeurs de la longueur du paquet IP, supérieure à 500 octets, on remarque que dans le cas de l'approche label-switching et quelque soit le nombre de spots, le *DO* est négligeable et représente moins que 1% de la longueur de la trame SEULE. Pour ces mêmes grandes valeurs de la longueur du paquet, l'approche self-switching révèle plus d'overhead que le label-switching sur la liaison montante du satellite avec une moyenne de 2.6%

en plus pour les 32 spots et environ 6.5% en plus pour les 82 spots. Pour les petites valeurs de la longueur du paquet IP (moins que 50 octets), le *DO* est plus grand que : 10% pour le cas de label-switching, 15% pour le cas de self-switching avec 32 spots, et 25% pour le cas de self-switching avec 82 spots.

Il est clair que le label-switching ajoute moins d'overhead que le self-switching sur la voie montante du satellite, avec environ 5 fois moins d'overhead pour le 32 spots et 10 fois moins pour le 82 spots. Elle réduit également la charge et la complexité au niveau des RCST (pas de calcul de label pour chaque segment), mais elle introduit plus de complexité au niveau du satellite.

L'approche appropriée (suitable approch), se décide au niveau du fournisseur des services IP par satellite en fonction de la stratégie adoptée : économique ou technique. Si l'opérateur (ou le fournisseur) doit assurer les services de la sécurité et de la livraison efficace de multicast avec un faible coût de *DO*, il devrait implémenter le SEULE avec le label-switching. Ce choix économique introduit plus de complexité au niveau du satellite. La deuxième option est de choisir le SEULE avec le self-switching qui est plus simple du point de vue technique surtout pour le satellite, mais elle introduit un coût (plus d'overhead) sur la voie montante du satellite et plus de complexité au niveau des RCST.

2.5 Conclusions

Dans ce chapitre nous avons proposé un nouveau système de sécurité basé chaos pour les communications IP multicast à travers le DVB-S. Ce système s'appuie sur une méthode d'encapsulation ULE améliorée (EULE) qui est capable de fonctionner avec les approches de commutation (label-switching et self-switching) afin d'assurer un transfert efficace de multicast atteignant seulement les spots beams contenant des membres.

L'EULE proposé requiert une légère modification sur le standard ULE et sur l'encapsulateur IP sans l'ajout d'aucun overhead. Il est adapté à la fois aux communications unicast et multicast. Pour sécuriser les trames EULE, une nouvelle version appelée SEULE satisfaisant toutes les exigences de sécurité est proposée. Elle ajoute seulement 6 octets d'entête d'extension à l'EULE, et elle authentifie et chiffre chaque paquet multicast avec une clé différente.

Les séquences chaotiques sont utilisées pour la génération des clés secrètes et pour le chiffrement des données et des clés transmises. Pour réduire la charge de rekeying sur toutes les ressources du système et surtout la liaison satellitaire, une architecture de deux couches LKH, TLKH est utilisée. De sorte que, le rekeying d'un membre affecte seulement le LKH terrestre et a un impact négligeable sur la liaison satellitaire. Le nombre des clés stockées au niveau de GKC est également réduit. En plus, nous introduisons l'utilisation de paquet KPDU qui transmet un ensemble de clés avec un petit entête et un mécanisme qui permet le rétablissement de la synchronisation.

Les simulations effectuées du système proposé et des systèmes flat, LKH et l'ECPVSS, ont montré la supériorité de notre système. En effet, le système proposé est capable de réduire à la fois le nombre des messages de rekeying, en moyenne, à 72% par rapport à la méthode LKH et la consommation de la bande passante des données de la gestion des clés transmises à plus de 58% par rapport à la meilleure approche existante ECPVSS.

Par ailleurs, les simulations ont montrés que le coût du système proposé pour assurer les services de sécurité et de livraison multicast, varie selon l'approche sélectionnée. Le label-switching est conservatrice de la bande passante avec 5 à 10 fois moins de taux de données ajoutées que le self-switching. Cependant, ce dernier conduit à plus de simplicité technique au niveau du satellite.

3. Sécurité dans les réseaux mobiles 3G et 4G

3. Sécurité dans les réseaux mobiles 3G et 4G

3.1 Introduction

Les générations des réseaux de téléphonie mobile ont évolué de manière progressive, avec environ une nouvelle génération tous les dix ans. En l'espace d'une vingtaine d'années, les réseaux de téléphonie mobile ont connu un essor remarquable et une évolution grandiose où les services offerts se sont multipliés, les terminaux se sont améliorés, les protocoles sont devenus plus complexes et sécurisés, et l'architecture du réseau est changée.

Actuellement, plus de 6 milliards d'abonnés [ITU, 2012] à travers le monde (soit 87% de la population mondiale) utilisent les nouveaux services de communications mobiles qui permettent aux utilisateurs de naviguer sur le web, consulter leurs courriers électroniques, télécharger les vidéos, de la musique, tout cela sur le même terminal et en mobilité.

La deuxième génération de réseaux mobiles (2G) a été introduite, au début des années 1990. Les deux systèmes 2G les plus connus sont l'IS-95 (Interim Standard 95) et le GSM (Global System for Mobile Communications) qui est le système le plus répandu. Une décennie plus tard, la troisième génération de réseaux mobiles (3G) est introduite au début du vingt et unième siècle. Elle regroupe deux familles de technologies ayant connu un succès commercial : l'UMTS (Universal Mobile Telecommunications System), issu du GSM et largement déployé autour du globe, et le CDMA 2000, issu d'IS-95 et déployé en Asie et en Amérique du Nord. L'idée principale qui a menée au système 3G, est la volonté des six organismes de normalisation (SDO, Standards Development Organizations) de l'Europe, l'Asie et l'Amérique du Nord, formant le 3GPP (3rd Generation Partnership Project), de définir une norme unique (qui n'est que l'UMTS) au niveau mondial et d'offrir une itinérance globale et internationale aux utilisateurs. Les objectifs de l'UMTS étaient d'accroître la capacité du système téléphonique pour le service voix mais surtout d'améliorer le support des services de données haut débit (services multimédia) et c'est l'UMTS après qui donnera naissance au 4G.

Encore une décennie s'est écoulée et le 3GPP a définit la quatrième génération des réseaux mobiles en 2008 dans la Release 8 [TS 22.278, 2008] [TS 24.301, 2008] [TS 36.300 2008]. Sa nouvelle technologie radio est connue sous l'acronyme «LTE» (Long Term Evolution) et le système complet est nommé «LTE/EPC», où «EPC» (Evolved Packet Core) représente le nouveau réseau cœur basé uniquement sur IP. Le terme technique pour le système LTE/EPC, que nous allons adopter, est « l'EPS » (Evolved Packet System). L'EPS/4G, conçue et optimisée pour la transmission de données, fournit : de très haut débits d'environ 100 Mb/s sur la voie descendante et 50 Mb/s sur la voie montante, des temps de réponse (latence) plus faibles, une qualité de service améliorée, la capacité d'interopérabilité avec les systèmes (2G, 3G), et un haut niveau de sécurité.

Avec les différentes applications intégrées en téléphonie mobile et les données sensibles touchant la vie privée des utilisateurs, la sécurité des systèmes 3G et 4G est déplacée de plus en plus à l'avant-garde de l'attention. Cette question de sécurité est indispensable pour assurer que le système fonctionne correctement, et pour protéger les abonnés et empêcher l'utilisation abusive du réseau tout en veillant des revenus pour l'opérateur du réseau mobile.

Ce chapitre aborde l'architecture et le mécanisme de sécurité des deux réseaux UMTS et l'EPS. En effet, nous sommes intéressés par la sécurité de l'EPS que nous comptons analyser et améliorer dans le prochain chapitre. Or, le réseau EPS n'a pas été conçu comme un système complètement nouveau, mais il a gardé les éléments essentiels d'architecture et de sécurité de l'UMTS en ajoutant des améliorations pour apporter plus de sécurité et robustesse. Dans la première partie de ce chapitre nous décrivons le fonctionnement, l'architecture, et les mécanismes de sécurité utilisés dans le réseau UMTS et dans la deuxième partie nous présentons l'architecture et la sécurité des réseaux EPS.

3.2 Architecture du réseau de troisième génération UMTS

Le réseau 3G est composé de trois secteurs principaux [Kaaranen *et al.*, 2005], représentés dans la figure 3.1, et qui sont les suivants :

- l'équipement utilisateur UE (User Equipement);
- le réseau d'accès appelé UTRAN (Universal Terrestrial Radio Access Networks);
- et le réseau cœur.

L'équipement utilisateur UE dispose d'une connexion radio pour accéder au réseau et à ses ressources radio à travers le réseau d'accès UTRAN. Ce dernier est relié au réseau cœur via différents interfaces. L'équipement utilisateur (UE) se compose de deux parties : l'équipement mobile ME (Mobile Equipement) et l'USIM (Universal Subscriber Identity Module). Le ME est l'appareil mobile qui contient la fonctionnalité radio et tous les protocoles nécessaires pour la communication avec le réseau. Il contient également l'interface utilisateur, y compris l'écran et le clavier. La carte USIM est une version améliorée de la carte SIM en 2G. Il s'agit d'une application stockée et exécutée sur une carte ou plateforme de type UICC (Universal Integrated Circuit Card) [TS 31.101, 2011] et il assure la sécurité du terminal et la confidentialité des communications. L'USIM contient toutes les données spécifiées par l'opérateur pour l'abonné, incluant les informations permanentes de sécurité (comme la clé secrète permanente K et les fonctions de sécurité qui s'utilisent dans la procédure d'authentification et d'établissement des clés).

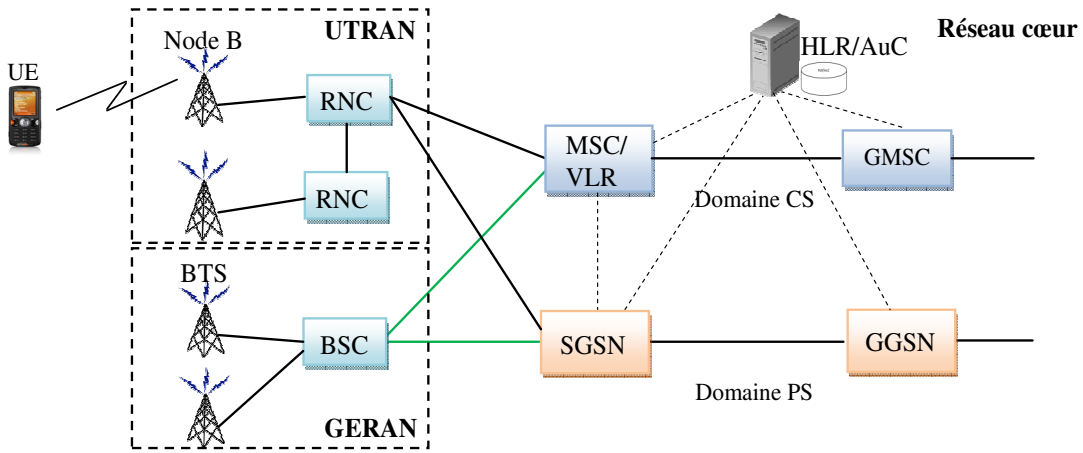

Figure 3.1. Architecture du réseau UMTS

L'UTRAN est composé de deux entités : la station de base Node B et le contrôleur de stations de base RNC (Radio Network Controller).

Le Node B est une antenne formée d'un ensemble d'émetteurs/récepteurs qui met l'interface radio à disposition de l'UE. Réparties géographiquement sur l'ensemble du territoire, les Nodes B sont au réseau UMTS ce que les BTS sont au réseau GSM. Ils gèrent la couche physique de l'interface radio. Il régit le codage du canal, l'entrelacement, l'adaptation du débit et l'étalement.

Le RNC est un contrôleur de Node B. Il contrôle et gère les ressources radio en utilisant le protocole RRC (Radio Ressource Control) pour définir les procédures de communication entre mobiles (par l'intermédiaire des Node B) et le réseau. Il gère : le contrôle de charge et de congestion des différents Node B ; et le contrôle d'admission et d'allocation des codes pour les nouveaux liens radio (entrée d'un mobile dans la zone de cellules gérées ...).

Le GERAN (GSM/EDGE Radio Access Network) est le réseau d'accès radio de 2G englobant la technologie radio GSM et son amélioration, EDGE (Enhanced Data rates for GSM Evolution). Elle est supportée dans 3G, où BTS (Base Transceiver Station) est la station de base en GSM et BSC (Base Station Controller) représente le contrôleur de BTS.

Le réseau cœur s'appuie sur les éléments de base du réseau GSM et GPRS. Il est en charge de la commutation et du routage des communications (voix et données) vers les réseaux externes. Il regroupe l'ensemble des équipements et des bases de données assurant les fonctions de l'enregistrement de l'UE au réseau et la mise à jour de sa localisation, le contrôle des appels, le contrôle de la sécurité, la gestion de l'interface avec les réseaux externes. Le réseau coeur se décompose en deux parties : le domaine circuit CS (Circuit Switched) et le domaine paquet PS (Packet Switched).

Le domaine circuit permettra de gérer les services temps réels dédiés aux conversations téléphoniques (vidéo-téléphonie, jeux vidéo, applications multimédia). Ces applications nécessitent un temps de transfert rapide. L'infrastructure s'appuiera alors sur les principaux éléments du réseau GSM : MSC (Mobile Switching Center) en charge du routage dans le réseau, de l'interconnexion avec les autres réseaux et de la coordination des appels, VLR (Visitor Location Register) est une base de données temporaire contenant des informations sur les utilisateurs se trouvant dans une certaine zone géographique, et le GMSC (Gateway MSC) qui permet la connexion directe avec les réseaux externes.

Le domaine paquet permettra de gérer les services non temps réels. II s'agit principalement de la navigation sur l'Internet, de la gestion de jeux en réseaux et de l'accès/utilisation des e-mails. Ces applications sont moins sensibles au temps de transfert, c'est la raison pour laquelle les données transiteront en mode paquet. Le débit du domaine paquet sera beaucoup plus rapide que le mode circuit. L'infrastructure s'appuiera alors sur les principaux éléments du réseau GPRS : SGSN (Serving GPRS Support Node) qui est une base de données existantes en mode paquet GPRS, et le GGSN (Gateway GPRS Support Node) est une passerelle d'interconnexion entre le réseau paquet mobile (GPRS ou UMTS) et les réseaux IP externes (Internet et les autres réseaux publics ou privés de transmission de données).

Les informations statiques de chaque abonné sont maintenues dans la base de données HLR (Home Location Register) comme son identité internationale, son numéro de téléphone et son profil. Il contient aussi le numéro du VLR où il existe. Le centre d'authentification AuC (Authentication Center) est habituellement intégré à un HLR qui stocke les informations de sécurité permanente liés aux abonnés (comme la clé secrète et les fonctions de sécurité). L'AuC génère des données d'authentification et de sécurité temporaire, qui peuvent être utilisées dans le réseau de service pour l'authentification de l'abonné et le chiffrement du trafic de l'utilisateur.

3.3 Sécurité dans la troisième génération (UMTS)

3.3.1 Principes et objectives de la sécurité de 3G

Le développement de la sécurité de l'UMTS a été réalisé par le groupe de travail de sécurité de la 3GPP, qui a définit dans [TS 33.120, 1999] les principes et les objectifs de la sécurité 3G. Les principes consistent à : garder les éléments robustes de la sécurité 2G, améliorer la sécurité 2G et ajouter de nouvelles fonctions de sécurité pour les nouveaux services 3G.

La sécurité UMTS a gardé les éléments de sécurité 2G suivant : l'utilisation des identités temporaires pour la confidentialité de l'identité, l'authentification des abonnés pour protéger l'accès aux services, et le chiffrement des données envoyées sur la voie radio pour la confidentialité des communications. Les principales améliorations (par rapport à 2G) sont les suivantes :

a) Le chiffrement des données a été étendu pour couvrir non seulement l'interface radio, mais aussi le lien entre le Node B et le RNC.

b) La taille de la clé de chiffrement a été augmentée de 64 bits à 128 bits, ce qui permet d'éviter l'attaque par recherche exhaustive.

c) L'authentification mutuelle est introduite afin de permettre aussi à l'utilisateur d'authentifier le réseau.

d) La protection de l'intégrité des données de signalisation est introduite pour authentifier les messages de contrôle échangés entre le mobile et le RNC.

e) La sécurité dans le réseau cœur est ajoutée afin d'éviter la transmission non sécurisé (en clair) des clés de chiffrement et des données d'authentification entre les différents éléments du réseau.

Ajoutons à tout cela, une liste d'objectifs pour sécuriser les nouveaux services 3G (commerce électronique, streaming vidéo, etc.).

3.3.2 Mécanismes de la sécurité 3G

3.3.2.1 Confidentialité de l'identité

Chaque abonné dispose d'une identité permanente internationale IMSI (International Mobile Subscriber Identity), unique et n'est connu qu'à l'intérieur du réseau mobile. Si l'utilisateur

envoie son identité permanente sur la voie radio pour s'identifier, un attaquant qui écoute le canal peut intercepter l'IMSI et par suite il peut suivre la localisation de l'abonné mobile en interceptant les messages échangés sur le canal radio. Pour cela il est essentiel d'assurer la confidentialité des identités des usagers. Ceci se réalise, en transmettant l'IMSI le plus rarement possible et seulement dans les cas nécessaires et en affectant à chaque abonné une autre identité temporaire TMSI (Temporary Mobile Subscriber Identity) qui sera utilisée à la place d'IMSI. En effet, l'identification de l'utilisateur dans l'UMTS est réalisée par deux identités temporaires et non pas par une seule [TS 43.020, 2011] et [TS 23.060, 2011] : on utilise le TMSI dans le domaine CS ou le P-TMSI (Packet TMSI) dans le domaine PS. Le premier enregistrement lorsque le téléphone est remis sous tension, l'IMSI est transmis puisqu'il n'y a pas encore de TMSI alloué. Ensuite, seuls les TMSIs successifs du mobile seront transmis sur la voie radio. Lorsqu'un utilisateur se déplace et veut utiliser le TMSI avec un réseau de service différent, ou dans une zone différente, il doit envoyer le LAI (Location Area Identity) ou le RAI (Routing Area Identity) correspondant au TMSI utilisé pour que le nouveau réseau de service puisse l'identifier.

Le TMSI est utilisé seulement entre l'utilisateur et le réseau de service. Quand le réseau de service fait référence à l'utilisateur dans le cadre des communications avec le réseau d'origine, il utilise l'IMSI. L'allocation d'un nouveau TMSI, TMSIn, est initiée par le VLR/SGSN. Le TMSIn est généré par le VLR et envoyé chiffrer à l'UE. L'UE répond avec un message de confirmation. Une fois le changement est réalisé, l'UE efface l'ancienne valeur du TMSI, le TMSIa, et le VLR efface l'association entre le TMSIa et l'IMSI de l'utilisateur (le TMSIa peut alors être utilisé par d'autres utilisateurs). Si cette procédure échoue, c.à.d. que l'utilisateur ne reçoit pas le TMSIn, ou le VLR perd le TMSIa, alors, dans ces cas, l'utilisateur doit s'identifier auprès du VLR avec son identité permanente, l'IMSI.

Si l'utilisateur change sa localisation et entre dans une zone contrôlée par un autre VLR, VLRn, le VLR d'où il vient, VLRa, doit, normalement, transférer l'IMSI de l'utilisateur avec les vecteurs d'authentification (qu'on n'a pas encore utilisé) au VLRn. Ceci se fait après une demande du VLRn contenant le TMSIa (reçu de l'UE). Le VLRn connait l'adresse de l'ancienne VLRa en se basant sur le numéro de LAI envoyé par l'UE avec son TMSIa. Si le transfert entre VLRn et VLRa n'est pas possible, l'utilisateur doit s'identifier auprès du VLRn avec son identité permanente. L'identification avec l'identité permanente, IMSI, est initiée par le VLR dans les deux cas décrits plus haut ou pour la première identification de l'utilisateur, lorsqu'il met en marche son portable. La réponse de l'utilisateur au message du VLR contient l'IMSI de l'utilisateur. Cette procédure est identifiée comme une faiblesse de sécurité de la norme UMTS.

3.3.2.2 Authentification et établissement des clés AKA (Authentication and Key Agreement)

Le protocole d'authentification et d'établissement des clés AKA est un protocole qui réalise l'authentification mutuelle entre l'utilisateur et le réseau UMTS. L'authentification de l'utilisateur permet au réseau de vérifier que l'identité (IMSI ou TMSI) transmise par le mobile est correcte afin de protéger l'opérateur contre l'utilisation frauduleuse de ses ressources, et d'autre part pour protéger les abonnés en interdisant à des tierces personnes d'utiliser leur compte. L'authentification du réseau permet à l'utilisateur de vérifier qu'il est connecté à un réseau légitime (réseau de service autorisé par le réseau d'origine). Cette procédure est basée sur une clé secrète permanente K de 128 bits, connue seulement par l'USIM et le centre d'authentification

AuC du réseau d'origine. La clé K n'est jamais transmise à travers le réseau, ni sur l'interface radio, ni entre les équipements fixes et l'abonné n'a aucune connaissance de sa clé secrète K. La procédure AKA est basée sur un ensemble de paramètres regroupés dans un vecteur d'authentification AV (Authentication Vector).

3.3.2.2.1 Vecteur d'authentification AV

Le vecteur AV est généré par le réseau d'origine et plus précisément par l'AuC à partir de six fonctions de sécurité (f0, f1,…f5). Puis il est envoyé au réseau de service VLR/SGSN dans le but d'effectuer la procédure AKA. Chaque vecteur AV est composé des cinq éléments suivants :

- RAND : une valeur aléatoire ;
- XRES : la réponse attendue par le réseau de service pour authentifier l'utilisateur ;
- CK : la clé de chiffrement ;
- IK : la clé d'intégrité ;
- AUTN : un jeton d'authentification utilisé par l'utilisateur pour authentifier le réseau.

Le paramètre RAND est une valeur aléatoire générée par la fonction f0. Il est envoyé en clair sur la voie radio et il est utilisé comme entrée pour toutes les autres fonctions de sécurité f1 à f5.

Comme le réseau de service ne connaît pas la clé sécrète K, alors pour authentifier l'utilisateur, le réseau d'origine envoie XRES (généré par la fonction f2) au réseau de service. Ce paramètre peut être généré seulement si on connaît la clé K. Si l'utilisateur envoie RES (RESponse) égale à XRES (Expected Response), alors l'authentification est réussie.

Les clés de chiffrement et d'intégrité CK et IK (Integrity Key) sont générés respectivement par les fonctions f3 et f4. Le VLR/SGSN reçoit ces clés dans l'AV émis du réseau d'origine (HLR/AuC). De son côté, l'USIM les génère au cours de la procédure AKA.

Le jeton d'authentification AUTN est la concaténation de trois paramètres ($SQN_{HE} \oplus AK$), AMF, et MAC, et il est envoyé par le réseau de service (VLR/SGSN) à l'utilisateur, pour que ce dernier puisse authentifier le réseau et avoir des informations sur certains aspects de la sécurité, comme par exemple les algorithmes de chiffrement utilisés. Pour chaque AV, un nouveau numéro de séquence SQN_{HE} (Sequence Number Home Environment) est généré par l'AuC à côté de la valeur aléatoire RAND.

L'AuC utilise un compteur de SQN_{HE} différent pour chaque utilisateur et cette valeur sera envoyée chiffrée par la clé d'anonymat AK (Anonymity key, généré par la fonction f5). L'USIM utilise une valeur seuil SQN_{MS} (Sequence Number Mobile Station) qui représente la valeur maximale du SQN_{HE} déjà acceptée par l'USIM. Si la valeur de SQN_{HE} reçue par l'USIM au cours de l'authentification est valide (dans la gamme correcte), l'authentification peut être réalisée avec succès. Sinon, l'USIM envoie une demande de resynchronisation.

Le deuxième champ de l'AUTN est le champ de gestion de l'authenticité AMF (Authentication and key Management Field). Ce dernier est utilisé pour envoyer des informations relatives aux algorithmes utilisés, pour modifier la valeur SQN_{MS} de l'USIM, ou pour déterminer la période de vie des clés CK et IK. Il peut être utilisé avec des objectifs différents par des réseaux différents. Le standard UMTS offre seulement un cadre général pour l'utilisation de l'AMF.

La valeur MAC = $f1_K(SQN_{HE}$, RAND, AMF) est la partie du jeton AUTN qui sera vérifiée par l'USIM pour authentifier le réseau. L'utilisateur fait confiance à son réseau d'origine. Si le réseau de service peut fournir un bon message MAC, alors l'utilisateur saura que son réseau d'origine a fait confiance au réseau de service et, par conséquence, l'utilisateur aussi.

3.3.2.2.2 Procédure UMTS-AKA

Cette procédure a comme double objectif, de réaliser une authentification mutuelle entre l'utilisateur et le réseau UMTS, et d'établir une nouvelle paire de clés de chiffrement et d'intégrité (CK, IK) commune entre l'USIM et le VLR/SGSN. La procédure UMTS-AKA, présentée dans la figure 3.2, commence lorsque l'utilisateur envoie son identité IMSI ou TMSI au réseau de service afin d'établir un appel ou pour transmettre des données.

Dans une première étape, le réseau de service demande au réseau d'origine les données d'authentification représentés par les vecteurs d'authentification AV(1…n). Cette demande doit inclure l'IMSI de l'utilisateur et le type du nœud demandeur (CS ou PS). Le HLR/AuC envoie alors un ensemble ordonné de n vecteurs d'authentification AV(1 .. n) classés selon le numéro de séquence.

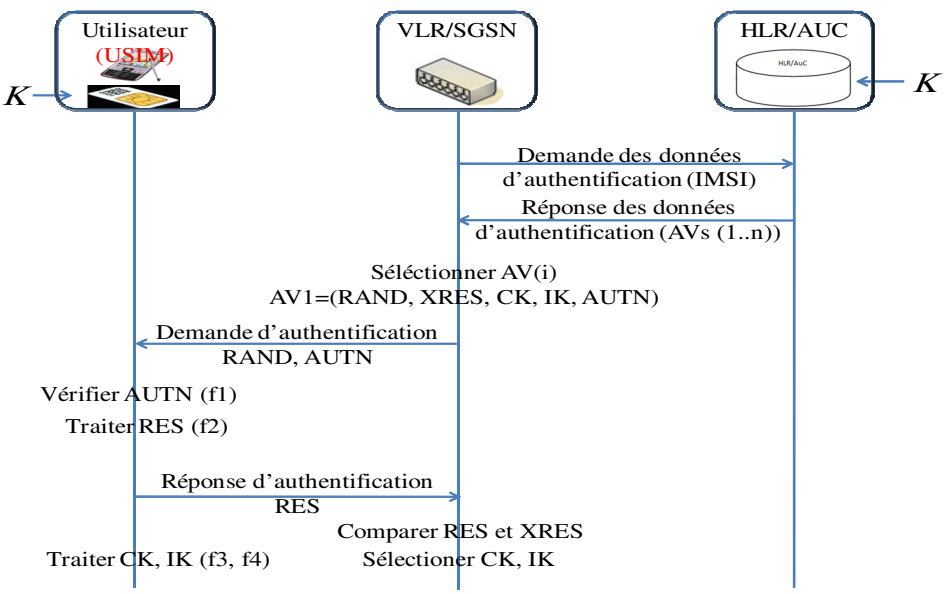

Figure 3.2. Procédure d'authentification et d'établissement de clés UMTS-AKA

Le réseau de service choisit un de ces vecteurs, AV(i), et l'utilise pour l'authentification. Un vecteur d'authentification peut être utilisé pour une seule authentification seulement.

Le réseau de service envoie à l'utilisateur une requête d'authentification (*User authentication request)* qui contient les paramètres RAND et AUTN de l'AV. L'USIM vérifie l'authenticité du jeton AUTN(i) par le calcul de XMAC=$f1_K(SQN_{HE}$, RAND, AMF) en utilisant la clé secrète K et les paramètres reçus RAND, AMF et SQN_{HE} déchiffré par AK. Il vérifie ainsi si la valeur calculée XMAC est égale à la valeur MAC contenue dans le jeton d'authentification AUTN reçu.

Si les valeurs MAC et XMAC sont égales, l'USIM vérifie alors la validité du numéro de la séquence, SQN_{HE}.

Si le SQN_{HE} est valide, le réseau est alors authentifié par l'utilisateur, et l'USIM calcule dans ce cas le paramètre RES=$f2_k$(RAND) et l'envoie dans le message réponse (*user authentication response)* au réseau de service. Ensuite il traite les clés de chiffrement et d'authenticité CK et IK et les envoie au ME qui va les utiliser pour le chiffrement et l'intégrité des données. Autrement, si le SQN_{HE} n'était pas valide, l'USIM envoie un message d'échec de la synchronisation (*synchronization failure*), qui va permettre au réseau de service de demander au réseau d'origine des nouveaux vecteurs d'authentification.

Quand le VLR/SGSN reçoit le message *user authentication response*, il vérifie si la valeur RES reçue est égale à la valeur XRES du AV utilisé. Si les deux valeurs sont égales, alors l'abonné est authentifié. Le VLR/SGSN sélectionne les clés CK et IK contenues dans l'AV et les envoie au RNC qui doit effectuer le chiffrement et l'intégrité des messages transmis.

Dans les deux cas où le XMAC est différent du MAC reçu, ou le RES reçu est différent du XRES, un message d'échec d'authentification (*Authentication Failure message*) qui inclue la raison de l'échec, est envoyé au réseau d'origine.

Une fois la procédure AKA est terminée, les clés établies entre l'USIM et le VLR/SGSN (CK, IK) seront envoyées aux équipements qui réalisent le chiffrement/déchiffrement et l'intégrité, c.à.d. le ME de l'utilisateur et le RNC du côté réseau. Ainsi ils peuvent commencer une communication sécurisée. Notons que les données usager sont seulement chiffrées alors que les messages de signalisation sont chiffrés et ont l'intégrité protégée.

3.3.2.2.3 Négociation des algorithmes et validité des clés CK, IK

Avant de commencer une communication sécurisée, les parties communicantes UE et RNC doivent se mettre d'accord sur les algorithmes de chiffrement UEA (UMTS Encryption Algorithm) et d'intégrité UIA (UMTS Integrity Algorithm) à utiliser. Pour cela, lorsque l'UE veut établir une connexion avec le réseau UMTS, il envoie au début un message indiquant les algorithmes UIA et UEA supportés par lui. Le réseau compare ces algorithmes avec ses propres algorithmes UIA et UEA, ses préférences et les éventuels requis spéciaux de l'abonnement. Ainsi, le réseau choisira un de ces algorithmes pour l'utiliser dans le cadre de cette connexion. Dans le cas où le réseau et l'UE n'ont aucun UIA et UEA en commun, et que le réseau n'est pas prêt à utiliser une connexion non chiffrée, la connexion sera libérée. Même si la procédure UMTS-AKA se déroule indépendamment pour les deux domaines, PS et CS, les algorithmes UIA et UEA négociés pour l'un de ces domaines sont utilisés par l'autre domaine.

D'autre part la procédure UMTS-AKA n'est pas utilisée chaque fois qu'un appel est initié, pour cela un mécanisme de contrôle de la période de validité des clés CK et IK est nécessaire afin d'assurer une protection contre les attaques avec des clés compromises. Pour cela, à chaque fois qu'une connexion se termine, deux paramètres représentant les valeurs des compteurs $START_{CS}$ et $START_{PS}$ sont envoyés par le réseau à l'UE. L'USIM compare ces valeurs reçues avec leur valeur maximale THRESHOLD fixée par l'opérateur. Si les valeurs reçues sont plus petites que THRESHOLD, leur valeur actuelle est stockée par l'USIM et les clés CK et IK actuelles restent valables. Sinon, l'UE va marquer la valeur actuelle du $START_{CS}$ et/ou $START_{PS}$ comme invalide,

effacera les clés actuelles CK et IK et mettra le paramètre KSI (Key Set Identifier) à '111' pour indiquer que les clés ne sont plus valides. Lorsqu'une nouvelle connexion RRC sera ouverte, l'USIM vérifie la validité de la valeur START. Si cette dernière est marquée comme invalide (c.à.d. égale ou supérieure à THRESHOLD) l'UE déclenchera une nouvelle procédure UMTS-AKA dans le domaine correspondant.

3.3.2.2.4 Procédure d'établissement du mode de sécurité

Décrivons maintenant la procédure qui mène à l'établissement d'une connexion sécurisée entre l'UE et le réseau. Les seules opérations permises, après l'envoie du «message L3 initial» (premier message de signalisation envoyé de l'UE au réseau de service, et qui peut être une demande d'attachement, une mise à jour de localisation,..) et avant le déclenchement du mode de sécurité, sont la procédure AKA et l'identification avec une identité permanente. La figure 3.3, montre les étapes qui conduisent à l'établissement d'une connexion sécurisée entre le ME et le RNC. Dans cette figure l'authentification, le début de la protection de l'intégrité et le début de chiffrement sont toujours présentées en ordre, sauf dans le cas où l'un de ces services n'est pas requis [TS 33.102, 2012] on le dépasse pour atteindre le suivant.

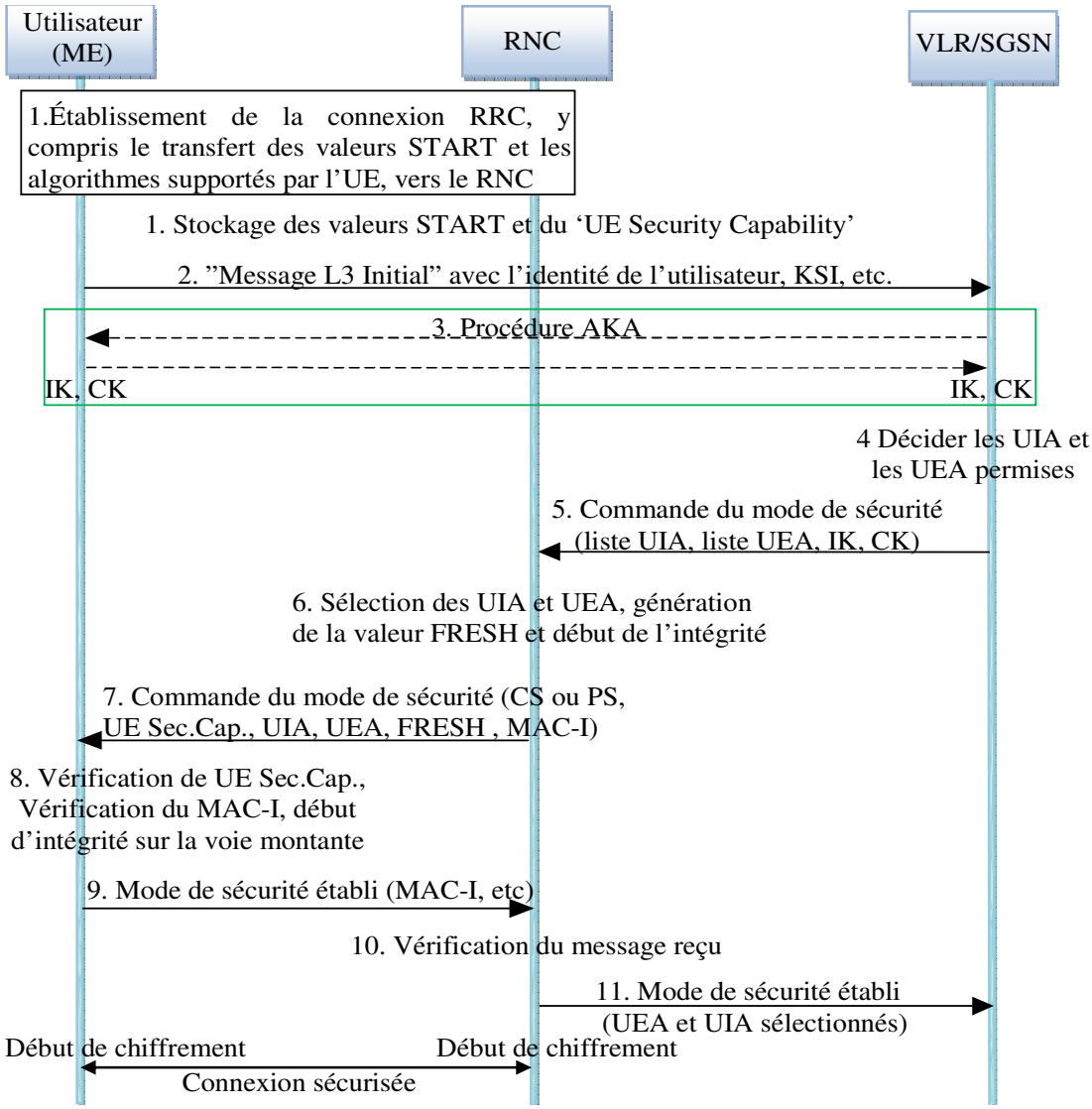

Figure 3.3. Authentification et établissement de la connexion sécurisée

Les différentes étapes de la figure 3.3 sont les suivantes :

1. Etablissement d'une connexion RRC entre l'UE et le RNC, incluant la transmission vers le RNC des valeurs START pour les deux domaines ($START_{PS}$ et $START_{CS}$), ainsi que les algorithmes de chiffrement et d'intégrité UIA et UEA supportés par l'UE (UE Security Capability).

2. L'utilisateur envoie le «message L3 initial» au VLR/SGSN. Ce message peut être une demande d'attachement (IMSI attach), une demande de mise à jour de localisation, une demande d'établissement d'appel, une réponse à un message d'avis de recherche (paging response) etc. Ce message contient l'identité de l'utilisateur et le KSI. Ce dernier est alloué par le domaine PS ou CS lors de la dernière authentification dans ce domaine, et il est utilisé pour permettre la réutilisation du CK et IK pendant l'établissement des connexions successives sans invoquer la procédure AKA.

3. Dans cette étape l'identité permanente de l'utilisateur peut être demandée. Si le KSI est égal à '111', il n'y a pas un ensemble de clés valides et la procédure AKA est lancée entre l'UE et le VLR/SGSN, ensuite une nouvelle KSI sera attribuée. Sinon, il y a une connexion sécurisée établie et les clés CK et IK seront utilisées.

4. Le VLR/SGSN détermine les UEA et UIA autorisés dans l'ordre de préférence.

5. Le VLR/SGSN utilise le message RANAP « commande du mode de sécurité » (security mode command) pour initier l'intégrité et le chiffrement. Ce message contient une liste ordonnée des algorithmes de sécurité (voir étape 4) et les clés IK et CK. Le message indique aussi si les clés CK et IK viennent d'être établies (voir étape 3, lorsque KSI=111) ou si elles ont été établies avant et ont déjà été utilisées. Si les clés ont été utilisées avant, la valeur du START envoyée dans la première étape est utilisée. Sinon, elle est mise à zéro.

6. Le RNC sélectionne les algorithmes qui seront utilisés. Il choisit le plus performant des algorithmes en fonction de la liste ordonnée reçue au point 5 et des algorithmes de sécurité supportés par l'UE, reçu au point 1. Le RNC génère une valeur aléatoire FRESH et déclenche la protection de l'intégrité sur la liaison descendante.

7. Le RNC génère le message RRC « commande du mode de sécurité » avec : la capacité de sécurité d'UE, les algorithmes qui seront utilisés (UIA et UEA), la valeur FRESH, la commande de début de chiffrement, le domaine du réseau cœur (CS ou PS) afin d'indiquer à l'utilisateur lequel des deux ensembles des clés doit utiliser, et le MAC d'intégrité de message ajouté à la fin de ce message. Ce message RRC représente le début de l'intégrité pour la liaison descendante.

8. A la réception du message RRC « commande du mode de sécurité », l'UE vérifie si la capacité de la sécurité (UE security capability) reçue est la même que celle envoyée dans le premier message, ensuite il vérifie l'intégrité du message reçu en calculant le XMAC-I en utilisant l'UIA et la valeur FRESH reçue.

9. Si toutes les vérifications sont correct, l'UE commence la procédure d'intégrité pour la voie montante, il envoie le message de confirmation « mode de sécurité établi » et lui attache le code MAC-I. Sinon, la procédure s'arrête là.

10. Le RNC vérifie l'intégrité du message reçu.

11. Si le message reçu est valide, le RNC envoie au VLR/SGSN le message RANAP « mode de sécurité établi » qui inclue les algorithmes sélectionnés et la procédure se termine ici.

À partir de cette étape toutes les données usager et de signalisation échangées entre l'UE et le RNC seront chiffrés par CK et la connexion sera sécurisée. En plus les messages de signalisation RRC seront authentifiés par IK. Par exemple si le message L3 initial était une demande d'attachement qui inclut l'IMSI de l'utilisateur et un KSI='111', après l'étape 11, le VLR/SGSN génère le TMSI pour l'envoyer à l'utilisateur à travers le RNC qui applique à son tour l'intégrité et le chiffrement sur ce TMSI avant de l'envoyer sur la voie radio.

3.3.2.2.5 Protection de l'intégrité et de la confidentialité

La protection de l'intégrité est effectuée pour la plupart des messages de contrôle et elle n'est pas utilisée pour la protection des données. Elle est réalisée par le calcul d'un code MAC-I qui est ajouté à la fin des messages, et ceci en utilisant la fonction d'intégrité f9 avec la clé IK. Le destinataire calcule le XMAC-I et le compare avec le MAC-I reçu. S'ils sont différents le message sera ignoré. Le chiffrement est effectué pour les messages de contrôle et pour les données aussi. Il est réalisé en utilisant la fonction de chiffrement f8 et la clé CK. L'algorithme de chiffrement par blocs KASUMI [TS 35.201, 2009] [TS 35.202, 2009] et l'algorithme de chiffrement par flux SNOW-3G [TS 35.215, 2012] [TS 35.216, 2012] sont les seules algorithmes définis par l'UMTS pour être utilisés comme des UIA et UEA. Notons que KASUMI a été cassé en 2005 par [Biham *et al.*, 2005]**.** En 2010 [Dunkelman *et al.*, 2010] ont publié une méthode de cryptanalyse facile à implémenter et qui permet de retrouver la clé de chiffrement de KASUMI. Donc les algorithmes UIA et UEA basés sur KASUMI ne sont plus considérés comme des algorithmes robustes.

3.4 Architecture du réseau de quatrième génération EPS

Le LTE (Long Term Evolution of 3G) a été envisagé dès novembre 2004 comme l'évolution à long terme de l'UMTS (d'où son nom). Il est considéré comme constituant une quatrième étape de l'évolution des réseaux d'accès mobiles, ou 4G. On peut ainsi véritablement parler d'une révolution de l'UMTS, plutôt que d'une évolution. Les objectifs essentiels du nouveau système consistent à assurer : des débits élevés aux utilisateurs, une faible latence, un haut niveau de sécurité, une amélioration de la qualité de service (QoS), et l'interfonctionnement avec les systèmes antérieurs (2G, 3G) [TS 22.278, 2012]. Pour répondre à ces objectifs, l'EPS a apporté une nouvelle interface radio et une nouvelle architecture plate basée sur IP (Internet Protocol).

En termes de vocabulaire, le futur réseau 4G s'appelle EPS (Evolved Packet System) ou LTE/EPC. Il est constitué d'un nouveau réseau d'accès appelé LTE, et d'un nouveau réseau cœur appelé EPC (Evolved Packet Core) ou SAE (System Architecture Evolution).

L'architecture du réseau EPS, définit dans les spécifications [TS 36.300, 2012] et [TS 23.401, 2012] et présentée dans la figure 3.4, supporte uniquement les services de données, où les services «circuit » migrent vers des services « paquet ». Elle fournit une connectivité IP entre l'UE et le réseau de paquets de données PDN (Packet Data Network), et elle permet le support des différents réseaux d'accès radio. Le service voix par exemple sera supporté par la voix sur IP VOIP (Voice Over IP).

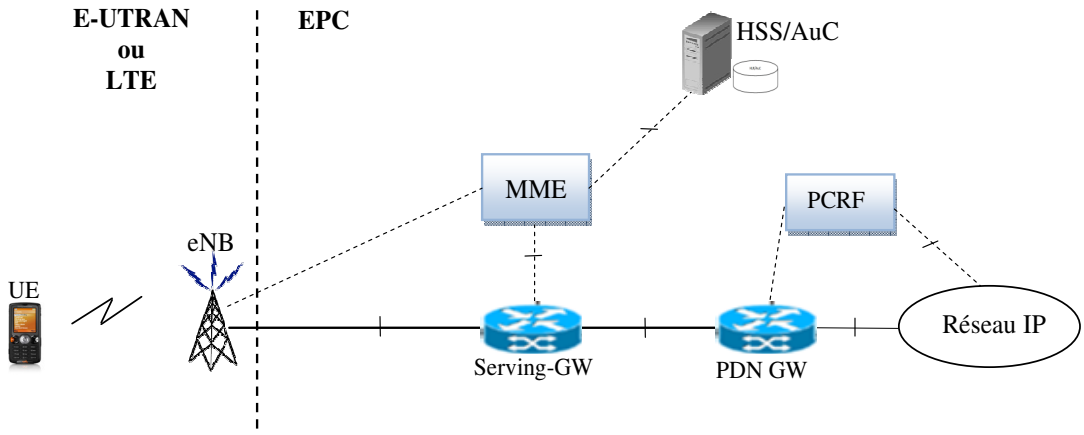

Figure 3.4. Architecture du réseau EPS

Dans la figure 3.4, on distingue deux types de trafic : le trafic usager représenté par des lignes continues; et le trafic de contrôle (signalisation) représenté par des lignes en pointillés.

Le réseau d'accès, LTE ou encore appelé E-UTRAN (Evolved UTRAN), est composé des nœuds eNB (evolved NodeB). L'EPC est composé de plusieurs nœuds qui sont : le MME (Mobility Management Entity), le HSS (Home Subscriber Server), le S-GW (Serving Gateway), le PDN GW (PDN Gateway) et le PCRF (Policy Control and Charging Rules Functions). Décrivons chacun de ces nœuds.

L'eNB est responsable de la transmission et de la réception radio avec l'UE. A la différence de l'UTRAN 3G où sont présentés les entités Node B et RNC, l'architecture E-UTRAN ne présente que des eNB afin de réduire la latence du système et rend l'architecture plus robuste et évolutive. Les fonctions supportées par le RNC ont été réparties entre l'eNB et les entités du réseau cœur MME/Serving GW. L'eNB peut gérer plusieurs cellules, et il est responsable de la gestion des ressources radio ou RRM (Radio Resource Management), telle que l'allocation dynamique des ressources à l'UE, le contrôle du radio *bearer*, l'exécution de *handover*, et la configuration des mesures de l'UE et de l'eNB. Il est également responsable de la sécurité des échanges sur l'interface radio où il effectue le chiffrement et la protection de l'intégrité ainsi que la dérivation des clés.

Le réseau de service MME est l'entité de gestion de mobilité qui gère la signalisation entre l'UE et le réseau cœur. Le MME est responsable de la gestion des *bearers* (établissement, reconfiguration et relâchement des *bearers*), de l'authentification des abonnées à partir des informations fournies par le HSS, de la dérivation des clés de sécurité, et de déclenchement de la sécurité.

Le HLR de l'architecture 3G est réutilisé et renommé Home Subscriber Server (HSS). Ce dernier est donc un HLR évolué qui contient les informations de souscription de l'utilisateur telles que le profil de l'abonné (IMSI, les paramètres de QoS autorisés, volume de données permise). Il contient aussi les informations concernant les réseaux de données (PDN) auxquels l'UE peut se connecter, ainsi que les données d'authentification. Par ailleurs, le HSS contient des informations dynamiques comme l'identité du MME auquel l'utilisateur est actuellement attaché. Le centre d'authentification AuC est éventuellement intégré au HSS.

Les passerelles de l'EPC, S-GW et PDN GW, servent principalement à diriger le trafic usager vers les PDN. La S-GW conserve des contextes sur les *bearers* de l'UE lorsqu'il est en veille. Lorsque la S-GW reçoit des données destinées à un UE en veille, elle contacte le MME pour notifier l'UE et rétablir ainsi les *bearers* associés. Le nœud S-GW sert aussi de point d'ancrage pour l'interfonctionnement avec les technologies d'accès 3GPP comme l'UMTS (UTRAN) et le GSM (GERAN). La PDN GW a pour rôle d'allouer de l'adresse IP à l'UE. Elle sert de point d'ancrage pour l'interfonctionnement avec les technologies d'accès non 3GPP telles que CDMA 2000 et WiMAX. Elle permet aussi de mettre en œuvre la facturation par flux de données, conformément aux règles définis par le PCRF. Ce dernier est un nœud optionnel permettant l'application des règles de gestion sur le trafic et la facturation de l'utilisateur en fonction de son offre.

3.5 Sécurité dans EPS

3.5.1 Principes de la sécurité 4G

La dernière version de la spécification décrivant tous les éléments de la sécurité de l'EPS est publiée en 2012 dans [TS 33.401, 2012]. Ce n'est qu'une amélioration de la sécurité UMTS où elle garde les éléments de sécurité robustes et nécessaires. Chaque mécanisme de sécurité 3G gardé, doit être adapté au nouveau contexte pour être appliqué à la nouvelle architecture de l'EPS. Ce dernier doit être également en mesure de fonctionner avec les systèmes antérieurs, de sorte que ces adaptations doivent être effectuées d'une manière rétro-compatible. En plus de ces éléments gardés et ces adaptations, de nombreuses nouvelles extensions et améliorations ont été introduites dans l'architecture de sécurité EPS.

Le niveau élevé de sécurité offert par EPS est assuré par l'introduction de nouveaux mécanismes, en particulier dans le domaine de l'accès au réseau. Parmi ces mécanismes, on cite : l'authentification du réseau de service introduit dans la procédure EPS-AKA; une nouvelle hiérarchie des clés ; et l'introduction de deux niveaux de sécurité. Nous allons décrire ces nouveaux mécanismes dans des paragraphes ultérieurs.

Les éléments robustes de la sécurité 3G qui ont été retenus dans la conception de l'architecture de sécurité 4G sont les suivants [Forsberg *et al.*, 2010]:

- Protection de l'identité permanente de l'utilisateur (confidentialité de l'IMSI) en utilisant des identités temporaires;
- Les éléments fondamentaux de la procédure UMTS-AKA
    1. Authentification mutuelle entre l'utilisateur et le réseau avec l'établissement de nouvelles clés de sécurité ;
    2. Calcul préalable des vecteurs d'authentification dans l'AuC ;
    3. Gestion des numéros de séquence ;
    4. Procédure de resynchronisation ;
    5. Fraîcheur des clés établies.
- Confidentialité des données usager et signalisation envoyées sur la voie radio ;
- La protection d'intégrité des données de signalisation ;
- Négociation des algorithmes de sécurité (de chiffrement et d'intégrité) ;
- Algorithmes de sécurité standardisés (en excluant le KASUMI) ;

- Sécurité dans le domaine réseau ;
- la possibilité d'amélioration des mécanismes de sécurité pour combattre des nouvelles menaces ou pour protéger des nouveaux services. On peut donc par exemple proposer et introduire de nouveaux algorithmes de sécurité (chiffrement ou autres);
- Utilisation de la carte USIM 3G comme module de sécurité (SIM de 2G ne fonctionne pas en 4G);
- la transparence pour l'utilisateur, c.à.d. l'utilisateur ne doit rien faire pour bénéficier des mécanismes de sécurité;

3.5.2 Exigences de la sécurité en EPS et les menaces principales

3.5.2.1 Exigences de la sécurité en EPS

Les exigences de sécurité de l'EPS ont été détaillées et publiées dans les deux documents [TS 22.278, 2012] et [TS 33.401, 2012]. Le premier fournit les exigences de sécurité générales (qu'on appelle de haut niveau et qu'on note H), tandis que le deuxième fournit les exigences de sécurité liées à la confidentialité (qu'on note C). Ces exigences peuvent être résumées comme suivants :

- **(H-1)** L'EPS doit assurer un haut niveau de sécurité et une grande robustesse.
- **(H-2)** Toute vulnérabilité de sécurité dans une technologie d'accès ne doit pas affecter les autres accès.
- **(H-3)** L'EPS doit fournir une protection contre tout type de menaces et tout type d'attaques.
- **(H-4)** L'EPS doit assurer l'authenticité des informations transmises entre le terminal et le réseau.
- **(H-5)** Des bonnes mesures de protection pour tout type de trafic écoulé doivent être fournies.
- **(H-6)** L'EPS doit s'assurer que les utilisateurs non autorisés ne peuvent pas établir des communications à travers le système.
- **(C-1)** L'EPS doit fournir pour tous les abonnés, différents types de confidentialité pour : les communications, la localisation, et l'identité.
- **(C-2)** La source, le destinataire, et le contenu des communications, doivent être bien protégés contre toute divulgation à des individus non autorisés.
- **(C-3)** L'EPS doit masquer les identités des utilisateurs afin de les protéger des personnes non autorisés.
- **(C-4)** L'EPS doit cacher la localisation de l'utilisateur, des gens non autorisés y compris l'utilisateur avec lequel il se communique.

Dans le paragraphe 3.5.4 nous allons voir comment toutes les fonctions de sécurité standardisées de l'architecture de sécurité EPS, répondent à la plupart de ces exigences.

3.5.2.2 Menaces contre EPS

La plupart des menaces de sécurité qui forment une source de préoccupation pour l'EPS, comme pour n'importe quel autre système antérieur, ont été publiées dans [TR 33.821, 2009], et elles peuvent être résumées comme suivant :

- *Menaces contre l'identité de l'utilisateur* (IMSI catching attack) : ce type de menace est très grave puisqu'il ouvre la voie à différentes attaques et puisqu'il est l'un des points faibles dans la sécurité de l'UMTS. Ces menaces sont déjà adressées par les exigences C-1 et C-3 ci-dessus.

- *Menaces de suivi d'UE* (Threats of UE tracking) : pour ce type de menaces on peut imaginer le suivi d'un utilisateur en se basant sur l'adresse IP liée à l'identité de l'utilisateur (IMSI ou TMSI), ou bien le suivi d'un utilisateur en se basant sur les messages de signalisation du transfert intercellulaire (handover).

- *Menaces liées à l'eNB* (Threats related to eNB) : comme par exemple la menace de compromettre physiquement la station de base eNB.

- *Menaces contre la manipulation des données de signalisation* (Threats against manipulation of control plane data) : les informations de contrôle peuvent être très utiles pour les attaquants afin de dévoiler l'identité de l'utilisateur. Ces menaces sont abordées par les exigences H-4, H-5 et H-6 ci-dessus.

- *Menaces d'accès non autorisé au réseau* (Threats of unauthorized access to the network) : un pirate non abonné peut utiliser les ressources du réseau dans ce cas. Ces menaces sont déjà abordées uniquement par l'exigence H-6.

- *Menaces liées à un déni de service* (Threats related to denial of service) : Le brouillage radio, ou le lancement d'une attaque distribuée à partir de plusieurs UE vers certaines parties du réseau, ou une attaque DoS (Denial of Service) contre un UE.

- *Menaces contre les protocoles radio* (Threats against the radio protocols) : Un attaquant compétent peut modifier les premiers messages d'établissement de connexion radio de l'UE.

La plupart de ces menaces ont été abordées par les exigences de haut niveau **H** et de confidentialité **C** mentionnées dans le paragraphe précédent. Les autres menaces comme celles liées à l'eNB sont adressées par d'autres exigences plus spécifiques. Cependant, la menace liée au déni de service (DoS) figure parmi les menaces les plus dures d'être empêchées. En effet, il est aussi difficile de trouver des mesures contre le brouillage radio, mais ce type d'attaque est simple à se faire détecté et attrapé. La source troublant le signal radio peut être localisée par des mesureurs du champ électromagnétique. Dans le chapitre suivant, nous allons proposer quelques solutions adéquates pour certaines de ces attaques citées ici dans ce paragraphe.

3.5.3 Architecture de la sécurité en EPS

3.5.3.1 Différentes domaines de sécurité

La mise en œuvre de la sécurité dans le système EPS suit une architecture similaire à celle du système UMTS. Elle est composée de 5 domaines d'application de la sécurité (appelé encore groupes des fonctions de sécurité), de telle manière que chaque domaine protège une partie du réseau contre certaines menaces, et accomplit certains objectifs de la sécurité. Ces domaines sont les suivants :

1. *La sécurité de l'accès au réseau* (Network access security), regroupe les fonctions de sécurité qui offrent à l'abonné un accès sécurisé aux services de l'EPS, et qui protège en particulier contre les attaques sur la liaison radio.

2. *La sécurité du domaine réseau* (Network domain security) est l'ensemble des fonctions de sécurité qui permettent aux nœuds du réseau d'échanger en toute sécurité leurs signalisations et les données utilisateur (par exemple par un tunnel IPsec entre eNB et MME) et de se protéger contre les attaques sur le réseau filaire.

3. La *sécurité du domaine utilisateur* (User domain security) est l'ensemble des fonctions de sécurité qui garantie un accès sécurisé de l'utilisateur au téléphone mobile.

4. La *sécurité du domaine applicatif* (Application domain security) est l'ensemble des fonctions de sécurité qui assurent la protection des messages entre l'application sur le terminal et le fournisseur de l'application.

5. La *visibilité et la configuration de la sécurité* (Visibility and configurability of security) est l'ensemble des fonctions informant l'utilisateur si une fonction de sécurité est en marche ou non, et si son accès aux services du fournisseur dépend de l'utilisation de la fonction de sécurité.

3.5.3.2. Sécurité de l'accès au réseau

Ce domaine est le plus important et le plus vulnérable parmi tous les cinq domaines puisqu'il assure la sécurité de la liaison radio, le maillon faible de tous les réseaux téléphoniques mobiles. La spécification technique [TS 33.401, 2012] de 3GPP définit quatre fonctions pour assurer la sécurité d'accès au réseau EPS: confidentialité de l'identité de l'usager et du terminal, authentification mutuelle entre l'UE et le réseau, confidentialité des données de l'usager et de la signalisation, intégrité des données de l'usager et de la signalisation. Avant de présenter chacune de ces fonctions il faut noter que la station de base eNB fait partie de l'E-UTRAN et par suite de l'accès au réseau. Sa sécurité est donc indispensable et nous allons la présenter aussi après la discussion des 4 fonctions du domaine.

3.5.3.2.1 Confidentialité de l'identité de l'usager et du terminal

Le but de cette fonction est d'empêcher les intrus qui écoutent le canal radio (eavesdroppers) d'obtenir des informations pour identifier les parties communicantes (protection contre les

attaques passives). Une fois qu'un appel ou une communication est établi, on identifie l'abonné par l'une des deux identités qu'on doit protéger et qui sont : l'identité de l'usager IMSI stockée dans la carte UICC; et l'identité du terminal qui vient en deux variantes IMEI (International Mobile Equipement Identity) ou IMEISV (IMEI and Software Version number), stockée dans le ME. Le standard EPS affirme que l'architecture de la sécurité offre les services suivants [TS 33.102, 2012], relatifs à la confidentialité de l'utilisateur :

- *Confidentialité de l'identité de l'utilisateur*: l'identité permanente de l'utilisateur (IMSI) ne peut pas être dévoilée par écoute sur la liaison radio;
- *Confidentialité de la localisation de l'utilisateur*: la position (la présence ou l'arrivé dans une zone) d'un utilisateur ne peut pas être déterminée par écoute sur le canal radio;
- *Non-traçabilité de l'utilisateur:* l'identité de l'utilisateur, auquel sont adressés les services de la voie radio, ne peut pas être connue pour un intrus.

Pour offrir ces services, l'utilisateur est en général, identifié avec des identifiants temporaires, (TMSI en UMTS) et GUTI (Globally Unique Temporary UE Identity) en EPS. Pour éviter la traçabilité de l'utilisateur, l'utilisateur ne doit pas être identifié pour une longue période de temps avec la même identité temporaire. En plus, tous les messages de signalisation transmis sur la liaison radio et qui peuvent révéler l'identité de l'utilisateur doivent être chiffrés.

Les améliorations apportées par l'EPS par rapport au 3G consistent à ne pas envoyer au réseau, l'identité du terminal (IMEI ou IMEISV) avant que les mesures de sécurité seront activées. Cette fonction répond aux exigences de la confidentialité C-1 et C-3 citées avant.

3.5.3.2.2 Authentification mutuelle entre l'UE et le réseau

Cette propriété garantit l'identité des correspondants où le réseau de service vérifie l'identité de l'utilisateur et ce dernier vérifie également l'identité du réseau d'origine ainsi que le réseau de service. Cette fonction s'est inspirée en grande partie de celle du 3G [TS 33.102, 2012] avec une amélioration introduite qui permet à l'UE la possibilité de vérifier l'identité du réseau de service auquel il est connecté. Dans l'authentification 3G et via l'UMTS-AKA, l'UE s'assure seulement qu'il est connecté à un réseau de service autorisé par son réseau d'origine pour le servir. L'authentification du réseau de service n'a pas été considérée nécessaire lorsque la sécurité 3G a été conçue puisqu'il y avait une hypothèse de confiance mutuelle entre tous les opérateurs UMTS. Cette hypothèse a été considérée comme invalide pour toute la durée de vie de l'EPS. Cette authentification mutuelle effectuée dans l'EPS répond donc aux exigences de sécurité H-2, H-4 et H-6.

3.5.3.2.3 Confidentialité des données de l'usager et de la signalisation

Le but de cette fonction est de chiffrer les données transmises entre l'UE et le réseau, et plus particulièrement sur l'interface radio, afin de les rendre incompréhensibles pour les écoutes clandestines (eavesdroppers). Le chiffrement des données usagers et des messages de signalisation RRC se fait entre l'UE et la station de base eNB (c'était entre l'UE et le RNC en 3G). Un nouveau mécanisme pour la protection de confidentialité de la signalisation NAS (Non-Access Stratum) entre l'UE et le réseau cœur (MME) est introduit par EPS. Le chiffrement de la signalisation et des données usager est recommandé par le 3GPP, mais laissé au choix de l'opérateur, comme en UMTS et en GSM.

En général, les protocoles concernant l'UE sont regroupés en deux strates (niveaux) principales : la strate d'accès (Access Stratum AS) contient les protocoles qui s'exécutent entre l'UE et le réseau d'accès c.à.d. entre l'UE et l'eNB ; tandis que la strate de non-accès NAS contient les protocoles qui s'exécutent entre l'UE et le réseau cœur indépendamment du réseau d'accès radio c.à.d. entre l'UE et le MME comme montre la figure 3.5. Notons que le protocole RRC est le seul protocole de signalisation au niveau AS.

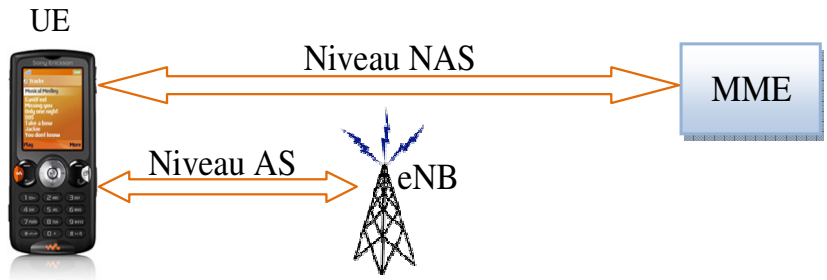

Figure 3.5. La strate d'accès AS et de non-accès NAS

Le chiffrement des messages RRC et des données usagers entre l'UE et l'eNB se réalise dans la couche PDCP (Packet Data Convergence Protocol), tandis que le chiffrement des messages de signalisation NAS entre l'UE et le MME se réalise dans la couche NAS EMM comme montre la figure 3.6. La confidentialité des données de l'usager et de la signalisation répond donc à l'exigence de haut niveau H-5 et aux exigences de confidentialité C-1 et C-2.

3.5.3.2.4 Intégrité des données de signalisation

Le système EPS affirme que tous les messages de signalisation NAS et RRC doivent être protégés en intégrité afin de vérifier l'authenticité de chaque message de signalisation et pour s'assurer que ces messages, NAS ou RRC, reçus n'ont pas été modifié en transit. Ce service offre aussi la protection contre la répétition (*replay protection)* où elle garantit qu'un message déjà reçu ne peut pas être réutilisé par un tiers ultérieurement. Ceci repose sur une synchronisation d'un compteur entre le terminal et le réseau. La protection d'intégrité de signalisation est obligatoire dans EPS à l'exception de certains messages échangés (entre l'UE et le réseau) avant l'activation de la sécurité, comme le message NAS *Attach Request* [TS 24.301, 2011], et le message *RRC Connection Request* [TS 36.331, 2011]. Cette protection et sa vérification sont réalisées par l'UE et l'eNB pour les messages RRC, et par l'UE et le MME pour les messages NAS. Cette fonction répond aux exigences de haut niveau H-4, H-5 et H-6.

Comme en 3G, le système EPS ne prévoit pas la protection d'intégrité pour les données usager, puisque le risque d'exploiter avec succès une modification sur les données chiffrées, est relativement faible, et que le taux des données (l'overhead) ajouté pour cette protection est significatif, surtout pour les services qui comptent sur des paquets de petite taille comme la voix.

Les mêmes couches mentionnées précédemment NAS et PDCP sont en charge d'assurer aussi l'intégrité des messages de signalisation NAS et RRC respectivement. La figure 3.6 montre ces couches au niveau de chaque équipement.

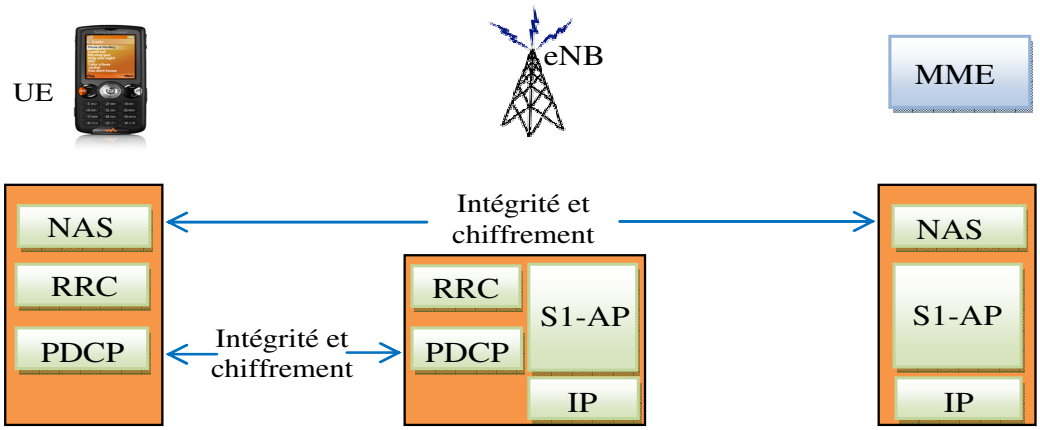

Figure 3.6. Protection de la signalisation NAS et RRC

3.5.3.3 La sécurité de l'eNB

L'importance de la sécurité de l'eNB vient du fait qu'il est un point de terminaison pour les mécanismes de sécurité principales de l'EPS, et qu'il est souvent installé dans des endroits exposés, en dehors du domaine de la sécurité de l'opérateur.

En effet, l'eNB est en générale situé sur un site distant non protégé et dont l'accès ne peut pas être complètement contrôlé par l'opérateur (toits d'immeubles, édifices publics, emplacements extérieurs,..). Il peut en être de même pour les liens physiques qui relient l'eNB (fibre optique enterrée, faisceau hertzien) à l'EPC.

C'est pour cela la norme 3GPP a établi pour la première fois pour un nœud d'un réseau téléphonique mobile des exigences de sécurité strictes pour l'eNB et ses interfaces [TS 33.401, 2012]. Ces exigences portent à la fois sur la sécurité logicielle (démarrage, configuration, gestion des clés, traitement des données) et sur les interfaces de l'eNB. Lorsque ces exigences sont satisfaites, l'eNB serait complètement sécurisé.

3.5.3.3.1 Démarrage et configuration de l'eNB

Les opérations de démarrage (boot) et de configuration des eNB doivent être authentifiées, afin d'empêcher les attaquants de modifier sa configuration via un accès local ou distant. Un exemple d'attaque locale, est qu'un attaquant peut avoir un accès physique à l'eNB et modifie certains éléments internes, ou utiliser une connexion directe à l'antenne de l'eNB et les interfaces du réseau pour intercepter ou injecter des données. C'est pour cela, les parties sensibles du processus du démarrage doivent être exécutées à l'aide d'un environnement sécurisé, et les transmissions et les implémentations de logiciels à l'eNB doivent être protégées en confidentialité et en intégrité.

3.5.3.3.2 Gestion des clés à l'intérieur de la station de base

Toutes les clés à l'intérieur de l'eNB doivent être protégées, et doivent se trouver dans un environnement sécurisé au sein de la plate-forme de cet eNB. Ces différentes clés sont utilisées pour assurer la confidentialité et la protection d'intégrité. En effet, l'EPC prévoit une clé de session spécifique pour chaque abonné (K_{eNB}). Cette clé donne naissance à 3 clés (comme nous

allons voir après) dont deux clés servent pour le chiffrement et pour l'intégrité de la signalisation RRC, et la troisième clé sert pour le chiffrement/déchiffrement des données de l'usager. L'eNB détient aussi des clés de longue durée et à long terme, utilisées pour s'authentifier à l'operateur du réseau, et des associations de sécurité utilisées pour la sécurité des interfaces terrestres. Toutes ces clés ne doivent jamais quitter l'environnement sécurisé.

3.5.3.3.3 Traitement des données du plan usager et de contrôle

La protection d'intégrité des messages RRC, ainsi que le chiffrement/déchiffrement des données usagers et de la signalisation RRC, doivent tous avoir lieu à l'intérieur de l'environnement sécurisé de l'eNB où les clés concernées sont stockées. La signalisation NAS n'est pas affectée par cette exigence, puisque l'eNB transmet les messages NAS protégés sans aucune interprétation. Idem pour le transport des données usager et de la signalisation sur les liens de l'eNB, il doit se faire aussi dans un environnement sécurisé et il doit être protégé par des moyennes cryptographiques.

3.5.3.4 Sécurité du domaine réseau

Cette fonction est héritée de celle de la 3G. Elle vise à protéger le trafic circulant entre les nœuds du réseau EPS (eNB, MME, S-GW,..), où elle assure : l'authentification mutuelle entre les nœuds communiquant, la confidentialité et l'intégrité des données. Comme le trafic au sein de ce domaine est totalement un trafic IP, donc on compte sur le protocole IPsec [RFC 4303, 2005] comme spécifié dans [TS 33.210, 2010], pour assurer la confidentialité, l'intégrité et la protection contre l'attaque par rejoue. Il est également nécessaire d'implémenter le composant d'IPsec, l'IKEv2 (Internet Key Exchange version 2) [RFC 4306, 2005] pour effectuer l'authentification mutuelle (en se basant sur des certificats), et pour établir les associations de sécurité (SA en anglais, pour Security Association). Ces associations de sécurité sont ainsi établies sur les interfaces entre l'eNB et MME/S-GW d'une part, et entre les eNB d'autre part. Les données de signalisation et usagers échangées sur ces interfaces doivent être protégées à l'aide du protocole IPsec ESP. En particulier, avant l'échange des messages de signalisation sur ces interfaces, une authentification basée sur des certificats et un échange de clés doivent se réaliser avec l'IKEv2. Les certificats et l'IKEv2 doivent être conformes au standard décrit par [TS 33.310, 2010].

Le mode tunnel d'IPsec doit être implémenté par l'eNB, et le mode transport d'IPSec peut être encore utilisé (facultatif) pour réduire la taille des données ajoutées sur les paquets par IPsec. L'utilisation d'IPsec/IKEv2 n'est pas obligatoire si les interfaces sont déjà protégées et considérées comme fiables (par exemple physiquement protégé).

3.5.3.5 Sécurité du domaine utilisateur

La sécurité du domaine utilisateur se porte sur deux aspects :

- *Authentification utilisateur – USIM* : cette authentification est réalisée avec un secret partagé par l'USIM et l'utilisateur autorisé (ou un groupe d'utilisateurs autorisés), e.g. le PIN, (Personal Identification Number). Elle a pour but de limiter l'accès à l'USIM et donc aux services de l'EPS, des utilisateurs non autorisés. Les mécanismes utilisés pour implémenter cette propriété sont décrits dans la norme [TS 31.101, 2011].

- *Authentification USIM – ME* : cette authentification est aussi réalisée avec des secrets partagés. Elle a pour but de limiter l'accès au ME pour les USIM qui ne sont pas autorisés. En pratique elle est utilisée par les opérateurs qui subventionnent le prix des mobiles afin de limiter l'utilisation de ces mobiles avec des USIM fournis par d'autres opérateurs.

3.5.3.6 Sécurité du domaine application

Cet aspect de la sécurité EPS permet aux réseaux EPS ou aux autres fournisseurs de services le développement des applications de sécurité sur la carte UICC. Ceci permet un transfert de données avec un degré de sécurité choisi par le réseau ou le fournisseur de l'application. La spécification technique qui permette le développement des applications sur l'UICC est décrite par [TS 31.111, 2009]. Les fonctions de sécurité pour ces applications sont décrites par [TS 23.048, 2005] et elles répondent aux besoins de sécurité identifiés par [TS 22.048, 2003].

3.5.3.7 Visibilité et configuration de la sécurité par l'utilisateur

Le but de cette fonction est d'informer l'utilisateur sur l'état de certaines fonctions de sécurité. Ces fonctions de sécurité doivent être en général transparentes pour l'utilisateur, et la norme 3GPP prévoit que l'utilisateur puisse être informé par son terminal de certains événements comme par exemple : 1) pour l'application de visibilité, il existe un indicateur de chiffrement (ciphering indicator) [TS 22.101, 2009] dans l'UE qui montre à l'utilisateur si la confidentialité des données est appliquée par le réseau ou non, en particulier lorsque le chiffrement des données est absent lors de l'établissement d'appel ; et 2) pour l'application de configuration, l'utilisateur a la possibilité d'activer ou désactiver l'authentification USIM-utilisateur réalisée par la saisie du code PIN.

3.5.3.8 Vue d'ensemble de la sécurité de l'EPS

Une vue d'ensemble des différents domaines de l'architecture de la sécurité EPS et les échanges entre ces éléments est donnée par la figure 3.7. Cette figure décompose l'architecture en trois niveaux, application, réseau et transport, qui sont les mêmes niveaux définis du modèle OSI. Les 5 domaines d'application de sécurité, que nous venons de définir, sont identifiés par les numéros sur les flèches reliant les différents blocs. C'est au domaine de l'accès au réseau (1) que nous allons concentrer nos efforts, puisque c'est la partie la plus spécifique à l'EPS et c'est la partie la plus fragile de n'importe quel réseau de télécommunication sans fil.

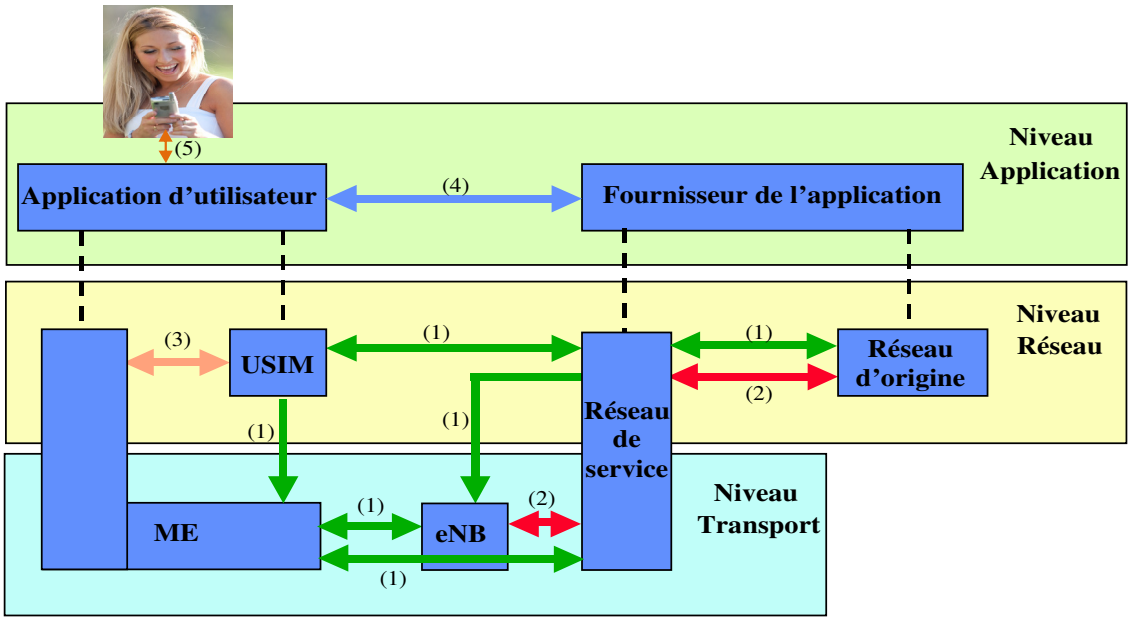

Figure 3.7. Vue d'ensemble des domaines de la sécurité dans le système EPS

La figure 3.8 montre l'application des fonctions de sécurité principales de l'accès au réseau et du domaine réseau entre les différents nœuds de l'architecture EPS. Lorsqu'un abonné demande l'établissement d'une connexion, pour faire un appel ou pour transmettre des données, il s'identifie (par son IMSI ou son GUTI), ensuite le MME demande des données d'authentification auprès du réseau d'origine. Ce dernier transmet les informations exigées, et puis le MME déclenche (avec l'UE) le protocole d'authentification et d'établissement de clés EPS-AKA (détaillé après). Une fois ce protocole est terminé avec succès, le MME et l'UE partagent une clé secrète appelée K_{ASME}, où l'acronyme ASME (Access Security Management Entity) réfère l'entité de gestion de la sécurité d'accès. Dans EPS, cette entité n'est que le MME.

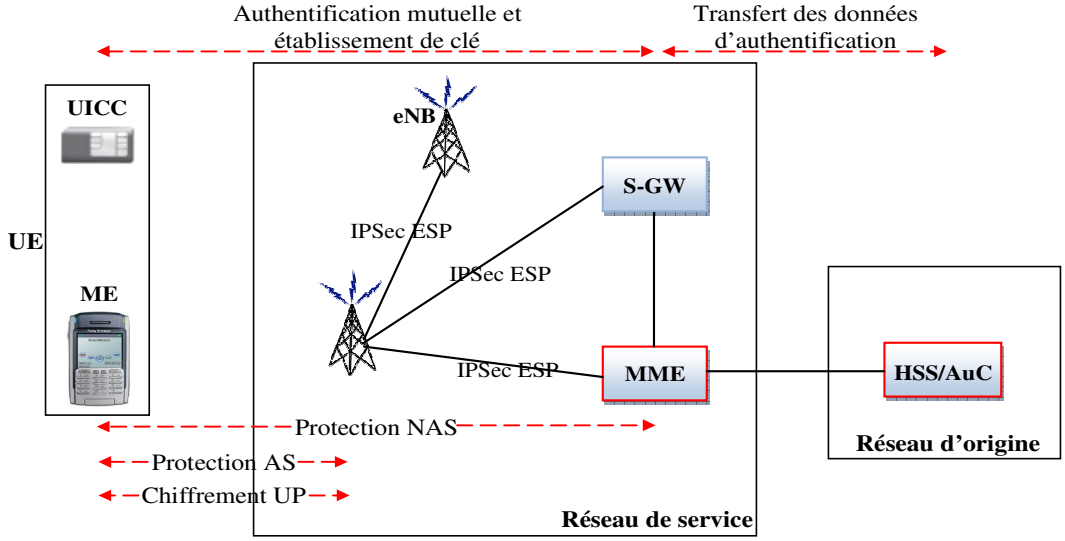

Figure 3.8. Architecture de la sécurité de l'EPS

Le MME et l'UE dérivent trois clés supplémentaires à partir de la clé K_{ASME}. Deux clés dérivées sont utilisées pour protéger la confidentialité et l'intégrité des données de signalisation entre le MME et l'UE. Ceci est représenté sur la figure 3.8 par la flèche 'Protection NAS'. La troisième clé dérivée est transmise à l'eNB dans le but de générer, avec l'UE, trois autres clés : deux clés, une pour la confidentialité et une autre pour l'intégrité des données de signalisation entre l'eNB et l'UE (représentées par la flèche notée par «protection AS»), et une troisième clé pour la confidentialité des données du plan usager entre l'eNB et l'UE (notée par la flèche «Chiffrement UP» (User plane)).

Notons que les données transportées sur les interfaces entre les différents nœuds du réseau de service (entre deux eNBs, ou entre eNB d'un part et MME ou S-GW d'un autre part) sont protégés, en confidentialité et en intégrité, par le protocole IPsec ESP (voir Figure 3.8) lorsque la protection cryptographique est appliquée.

Les fonctions de sécurité offertes par l'architecture de sécurité, et comme nous avons vu, répondent à la plupart des exigences de sécurité (de haut niveau H et de confidentialité C). Les exigences qui ne sont pas respectées, et que nous allons traiter dans le chapitre suivant, sont : l'exigence H-3 (où l'EPS doit assurer la protection contre les menaces et les attaques) ; l'exigence C-4 ; et les exigences C-1 et C-3 sont partiellement respectées (contre les écoutes mais pas contre les attaques actives (IMSI catcher)).

3.5.4 Accès sécurisé au réseau EPS

Dans cette section nous expliquons comment les utilisateurs sont identifiés et authentifiés par le réseau EPS afin de leur permettre d'accéder à ses ressources. Pour ceci, nous définissons les paramètres qui permettent d'identifier les terminaux et les abonnés et comment on les protège. Ensuite nous présentons le protocole EPS-AKA, ainsi que la génération des clés multiples.

3.5.4.1 Identification des abonnés et des terminaux

Comme en GSM et en 3G et pour des raisons de sécurité, l'EPS doit utiliser l'identité permanente IMSI pour identifier un abonné le minimum possible. Pour cette raison, l'EPS compte sur deux identités temporaires associées : le GUTI (Globally Unique Temporary Identity) et le C-RNTI (Cell Radio Network Temporary Identity). Le GUTI est l'identité temporaire allouée à l'UE par l'EPS afin qu'il n'ait pas à dévoiler son IMSI, tandis que le C-RNTI [TS 36.331, 2011] est affecté et utilisé lors du transfert intercellulaire (handover).

Le GUTI est un peu différent dans sa structure que TMSI et elle est composée de deux éléments principaux :

- le GUMMEI (Globally Unique MME Identifier), qui identifie le MME qui a alloué le GUTI. C'est une identité internationale et unique du MME. Il est construit à partir du MCC (Mobile Country Code), MNC (Mobile Network Code) et de l'identificateur de MME (MME Identifier) ;
- le M-TMSI, qui identifie uniquement l'UE dans le MME (qui a alloué le GUTI).

Le MME attribue un GUTI à l'UE après une demande d'attachement (*Attach request*) ou après une demande d'une mise à jour de zone de suivi, TAU (*Tracking Area Update request*). Le MME peut également attribuer un GUTI dans des procédures séparées de réallocations de GUTI [TS

23.401, 2012]. Dans chacun de ces cas, le MME n'envoie le GUTI qu'après l'activation de la protection de la signalisation NAS (nous revenons à ce point après).

Si le réseau supporte la confidentialité de signalisation alors un attaquant qui écoute les données transmises sur le lien entre le MME et l'UE ne peut pas lire le GUTI, et par suite il ne peut pas associer le GUTI à l'IMSI (ou à un ancien GUTI). Ce mécanisme protège l'identité temporaire de l'utilisateur contre les attaques passives (espionnage). Il empêche aussi le suivi d'un utilisateur (tracking a user) en observant d'une façon consécutive les GUTI attribuées à un même utilisateur.

En temps normal, l'UE est identifié à l'aide du GUTI. Cependant, si le MME ne peut retrouver l'IMSI associé au GUTI indiqué par l'UE, il demande à ce dernier de fournir son IMSI par une demande d'identité (contenant le GUTI). De même, lorsque l'UE ne dispose pas de GUTI valable (lors du premier enregistrement), il fournit son IMSI dans le premier message EMM (EPS Mobility Management).

D'un autre côté, chaque téléphone mobile est identifié physiquement par un numéro stocké dans une mémoire non volatile de l'équipement mobile. Ce numéro, appelé IMEI, contient une identification du type d'équipement mobile et un numéro de série servant à identifier de manière unique un équipement mobile donné. De plus, un équipement mobile est caractérisé par une version de logiciel SVN (Software Version Number) indiquant l'état de mise à jour du logiciel de base installé sur l'équipement mobile. La combinaison de l'identification du type et du numéro de série de l'équipement mobile avec la version de logiciel (SVN) donne une nouvelle identification, appelée IMEISV.

L'EPS a apporté une amélioration par rapport au 3G en ce qui concerne la confidentialité du terminal. Dans 3G, il est possible que le réseau demande l'identité du terminal à n'importe quel moment, même avant que la protection de signalisation est établie, et l'UE répond dans ce cas en envoyant l'identité de son terminal en clair. Comme un utilisateur tend à utiliser le même terminal pendant une longue durée, l'identité du terminal peut donner des indications fortes sur l'identité de l'utilisateur. Ceci n'est plus possible en EPS, puisque l'UE ne doit jamais envoyer son IMEI ou son IMEISV au réseau avant que la sécurité NAS est activée (même si le réseau demande).

3.5.4.2 Authentification et établissement des clés EPS-AKA

La procédure EPS-AKA est similaire à l'UMTS-AKA et elle vise les mêmes objectifs énumérés dans le paragraphe 3.5.1. Avec quelques légères améliorations introduites. La première amélioration consiste à générer des clés multiples à partir de la clé établie et obtenue par l'EPS-AKA, pour former une hiérarchie de clés. La deuxième amélioration consiste dans le fait que l'EPS-AKA assure l'authentification implicite du réseau de service ainsi que le réseau d'origine.

En effet, l'EPS-AKA, et comme l'UMTS-AKA, se base sur une clé secrète permanente K partagée entre l'USIM et l'AuC, et se divise en trois étapes :

- sur demande du MME, génération par le HSS des vecteurs d'authentification AV, et transmission de ces vecteurs au MME ;

- authentification mutuelle et établissement d'une clé commune K_{ASME} entre le réseau de service et l'UE;

- distribution des données d'authentification à l'intérieur et entre les réseaux de service.

Dans EPS-AKA, le ME, le HSS et le MME jouent des rôles similaires à leurs homologues (ME, HLR, et VLR/SGSN) dans l'UMTS-AKA avec quelques fonctionnalités de plus comme par exemple la dérivation des clés multiples à partir de la clé principale K_{ASME}.

3.5.4.2.1 Génération des vecteurs d'authentification EPS

Le vecteur d'authentification AV dans EPS est basé sur celui de l'UMTS avec quelques légères modifications sur certains paramètres. Nous avons vu que le vecteur AV en UMTS est composé des 5 éléments suivants: un nombre aléatoire RAND, une réponse attendue par le réseau XRES, une clé de chiffrement CK, une clé d'intégrité IK et un jeton d'authentification AUTN. Lorsque l'EPS-AKA est déclenché, l'AuC génère exactement les mêmes vecteurs d'authentification AV de l'UMTS à partir des six fonctions de sécurité (f0, f1,.., f5) et il les envoie au HSS. Ce dernier génère à partir des deux clés CK et IK, et à partir de ($SQN_{HE} \oplus AK$) et de l'identité du réseau de service SN id (Serving Network identity), une clé maîtresse K_{ASME}. Ceci est réalisé en appliquant la fonction de dérivation de clé KDF (Key Derivation Function). Les clés CK et IK utilisées pour calculer le vecteur d'authentification EPS ne doivent jamais quitter le HSS et elles sont supprimées juste après l'obtention de la nouvelle clé K_{ASME}. Le vecteur d'authentification EPS, est alors donné par AV EPS = (RAND||XRES||K_{ASME}||AUTN).

La figure 3.9 montre la génération d'un vecteur d'authentification AV UMTS (par l'AuC), et la génération d'un vecteur AV EPS (par le HSS) à partir de l'AV UMTS.

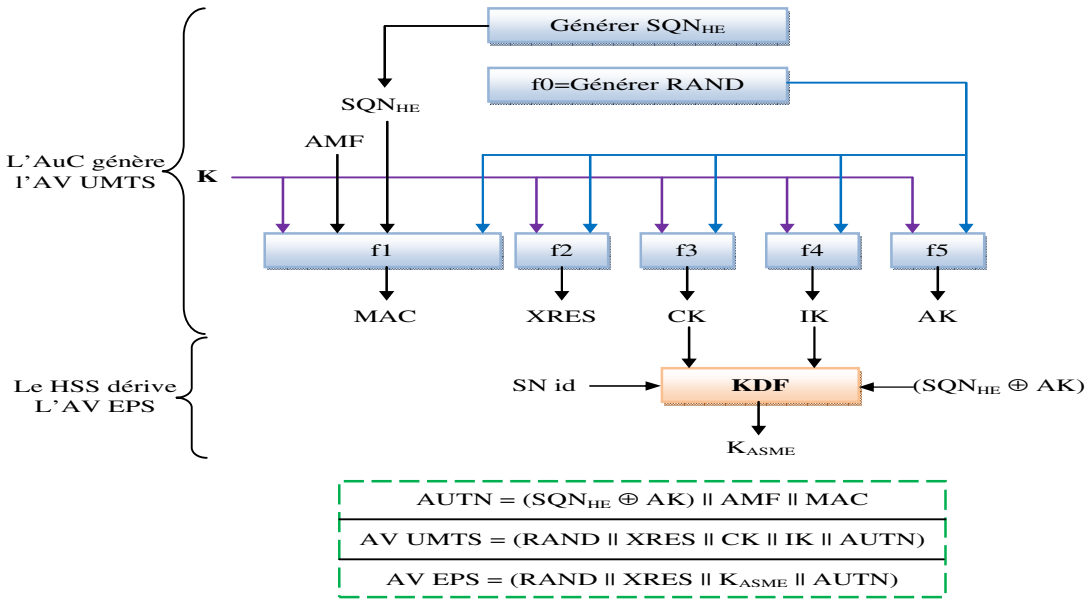

Figure 3.9. Génération du vecteur d'authentification AV EPS

Rappelons que le jeton AUTN, utilisé par l'utilisateur pour authentifier le réseau, est donné par AUTN = ($SQN_{HE} \oplus AK$|| AMF|| MAC). Si l'opérateur décide que le masquage de SQN_{HE} n'est pas nécessaire, alors il annule la fonction f5 pour rendre AK = $f5_K$ (RAND)=0.

Un numéro de séquence SQN_{HE} frais et une nouvelle valeur aléatoire RAND sont générés par l'AuC après une demande du HSS. Le RAND est généré par la fonction f0 et il est utilisé comme

entrée pour toutes les autres fonctions f1 à f5. La génération du SQN_{HE} est basée sur : un compteur incrémenté pas à pas, ou sur le temps, ou également sur la combinaison de ces deux ensembles [TS 33.102, 2012]. La prédictibilité du SQN_{HE} peut compromettre l'identité et la localisation des utilisateurs, pour cela, sa valeur est envoyée souvent chiffrée avec la clé d'anonymat AK (Anonymity Key). L'USIM utilise une valeur seuil SQN_{MS} qui représente la valeur maximale du SQN_{HE} acceptée par l'USIM dans la dernière authentification. Si la valeur de SQN_{HE} reçue par l'USIM au cours de l'authentification est valide (dans la gamme correcte), l'authentification peut être réalisée avec succès. Sinon, l'USIM envoie une demande de resynchronisation.

Le champ AMF inclut dans l'AUTN joue dans EPS, en plus de son rôle précédent joué dans UMTS, un nouveau rôle important. Il s'agit de distinguer entre les vecteurs d'authentification destinés pour 'l'usage EPS' de ceux qui sont destinés pour 'l'usage UMTS' (ou legacy uses). Le 3GPP a définit le bit le plus significatif (appelé 'bit de séparation AMF') des 16 bits de l'AMF pour cet objectif. L'AuC met ce bit à 1 dans les vecteurs à 'usage EPS', et à '0' sinon. C'est seulement l'AuC qui peut modifier les bits du champ AMF, et c'est le HSS qui l'informe de l'usage du vecteur lorsqu'il fait la demande les vecteurs d'authentification.

Les différentes tailles de l'ensemble des paramètres utilisés et générés au cours de la procédure AKA sont données dans le tableau suivant.

Paramètre	Longueur (bits)
K	128
RAND	128
XRES, RES	32-128
CK	128
IK	128
K_{ASME}	256
AUTN	128
SQN	48
AK	48
AMF	16
MAC, XMAC	64
SN id	20

Tableau 3.1. Longueur des paramètres d'authentification

3.5.4.2.2 Procédure EPS-AKA

Le but de cette procédure est de réaliser une authentification mutuelle entre l'utilisateur et le réseau et d'établir une nouvelle clé maîtresse K_{ASME} commune entre l'UE et le MME. Pendant l'authentification, l'USIM vérifie la fraîcheur du vecteur d'authentification (via la vérification du SQN_{HE} reçu) et authentifie son origine (HSS). La clé K_{ASME} va servir après à dériver des clés multiples utilisées dans les procédures de protection de la signalisation NAS, de la signalisation RRC, et du plan usager.

Le déroulement de la procédure EPS-AKA est présenté dans la figure 3.10. Cette procédure est initiée par le MME après l'identification de l'utilisateur par son IMSI ou GUTI. Dans une

première étape le MME demande au HSS les vecteurs d'authentification AV EPS. Cette demande, appelée *Authentication Data Request*, doit inclure : l'IMSI, l'identité du réseau de service 'SN id' du MME demandeur, et une indication que les données d'authentification sont demandées pour l'EPS (et non pas pour l'UMTS). Dans le HSS et comme nous venons de voir, le SN id entre dans le calcul de K_{ASME}.

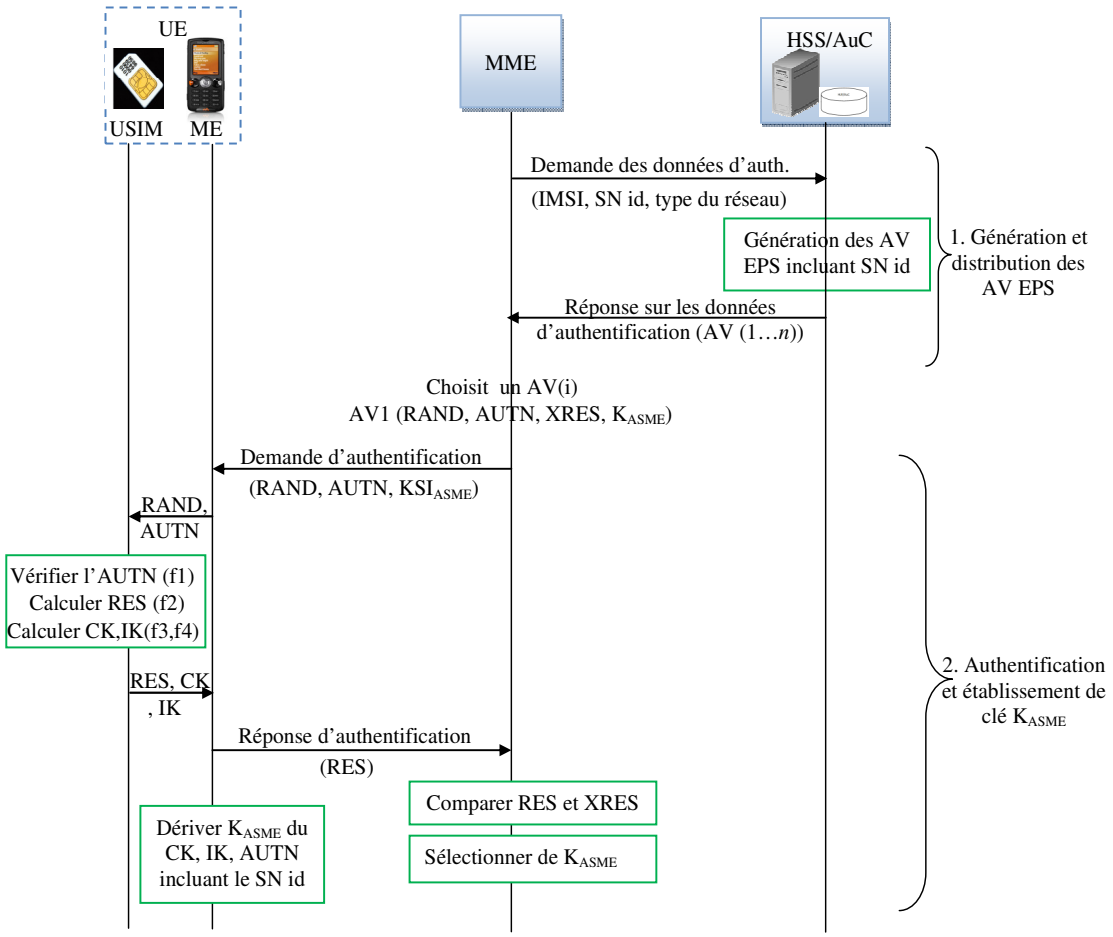

Figure 3.10. Procédure d'authentification et d'établissement de clé en EPS (EPS-AKA)

À la réception de la demande du MME, le HSS calcule les vecteurs AVs (ou il les cherche dans sa base de données s'ils sont déjà pré-calculés) et il les envoie au MME dans sa réponse appelée *Authentication Data Response.* Cette dernière contient un ensemble ordonné de *n* vecteurs d'authentification AV EPS classés selon le numéro de séquence, AV (1 ... *n*).

Ensuite, le MME choisit un de ces vecteurs, AV (i), et il l'utilise pour effectuer l'authentification. Un vecteur AV EPS peut être utilisé pour une seule authentification. Le MME envoie après à l'équipement utilisateur UE, et plus précisément à l'équipement mobile ME qui le transmet à son tour à l'USIM, une demande d'authentification (appelée *User Authentication Request*) contenant les paramètres RAND et AUTN extrait du vecteur AV(i) choisit. Le MME envoie également et

dans cette même demande d'authentification, un identificateur de la clé K_{ASME}, KSI_{ASME} (ou appelé eKSI, evolved KSI).

Dès réception des deux paramètres RAND et AUTN, l'USIM effectue le calcul des différents paramètres comme indiqué dans la figure 3.11. Au début l'USIM détermine la clé AK = $f5_K$ (RAND) qui lui permet le déchiffrement du numéro de séquence SQN = ($SQN_{HE} \oplus AK) \oplus AK$. Ensuite, l'USIM utilise le SQN calculé, le champ AMF inclut dans l'AUTN reçu, et le RAND reçu pour calculer la valeur XMAC=$f1_K$(SQN || RAND || AMF) dans le but de vérifier l'authenticité du jeton. Si XMAC est égale au MAC reçu inclut dans l'AUTN, l'USIM confirme la validité du SQN calculé et vérifie s'il est dans la gamme correcte ou non. Si oui, le réseau est alors authentifié. Si non, l'utilisateur envoie soit un message d'échec d'authentification en expliquant la raison de l'échec, soit un message d'échec de synchronisation (*synchronization failure*) à cause du SQN invalide (voir l'annexe B1 pour plus de détails).

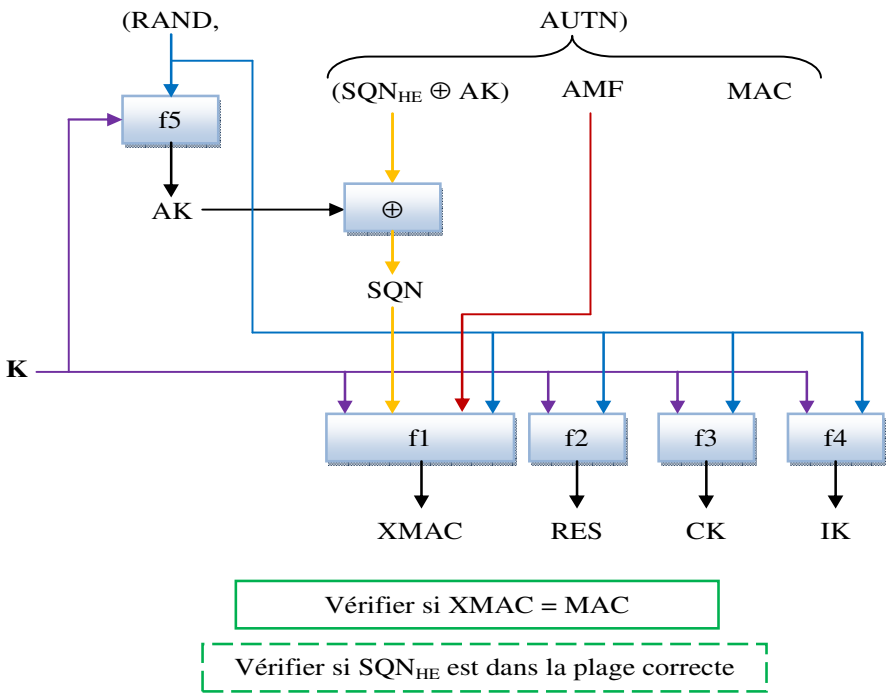

Figure 3.11. Fonctions d'authentification du réseau effectué par l'USIM

Si le SQN_{HE} est vérifié qu'il se trouve dans la gamme correcte, l'USIM calcule RES = $f2_K$(RAND) ainsi que les clés CK = $f3_K$ (RAND) et IK = $f4_K$ (RAND), et il les envoie au ME (nous sommes toujours dans UE). Ce dernier calcule en cas où il a reçu le bit de séparation d'AMF égale à 1 (usage EPS) la clé K_{ASME} en utilisant la même fonction KDF et les mêmes paramètres d'entrée utilisés par le HSS. Après, le ME supprime tout de suite les clés CK et IK, puis il envoie le RES dans un message de réponse d'authentification (*User Authentication Response*) au MME.

Le MME vérifie ainsi si le RES reçu correspond à la réponse attendue XRES inclut dans l'AV (i). Si c'est le cas, alors l'utilisateur est authentifié avec succès, et le MME sélectionne la clé K_{ASME}

contenue dans AV(i) pour l'utiliser dans les étapes suivantes. Sinon, c.à.d. si le MME trouve que XRES est différent de RES, alors il décide soit initier une nouvelle procédure EPS-AKA, soit abandonner la procédure d'authentification et envoyer un message de rejet d'authentification (*Authentication Reject message*) à l'UE [TS 24.301, 2011].

En EPS, un ME qui accède l'E-UTRAN doit toujours vérifier durant l'authentification que le bit de séparation AMF est mis à '1' par l'AuC. Cette vérification donne également au ME l'ordre de dériver la clé K_{ASME} dès la réception des clés CK, IK de l'USIM.

3.5.4.2.3 Distribution des données d'authentification à l'intérieur et entre les réseaux de service

Lorsqu'un utilisateur UE se déplace, le MME qui lui rend service peut changer. Dans ce cas, lorsque l'UE envoie une demande d'attachement (Attach Request), ou une demande de mise à jour de zone de suivi (Tracking Area Update Request) [TS 23.401, 2012], il va utiliser son identité temporaire GUTI, afin de garder son IMSI caché. Or le nouveau MME n'est pas en mesure de comprendre le GUTI, et il se trouve donc devant deux possibilités: soit il demande l'IMSI de l'UE et ce n'est pas du tout conseillé pour des raisons de sécurité, soit il demande de l'ancien MME avec qui l'UE était attaché (et qui a délivré le GUTI à l'UE), de traduire le GUTI en IMSI. Dans ce dernier cas, l'ancien MME doit également envoyer les données d'authentification au nouveau MME à côté de l'IMSI. Le type des données d'authentification qui doivent être échangées entre l'ancien et le nouveau MME diffère selon l'emplacement des deux MME, s'ils résident dans le même réseau de service ou dans des réseaux différents.

1- Lorsque les deux MME résident dans le même réseau de service, alors tout contexte de sécurité et tous les vecteurs d'authentification AV qui se trouvent dans l'ancien MME et qui ne sont pas encore utilisés, sont transmis avec l'IMSI au nouveau MME. Ce dernier peut utiliser les vecteurs d'authentification reçus puisque les deux MME ont le même SN id.

2- Lorsque les deux MME résident dans des réseaux de service différents, les vecteurs inutilisés dans l'ancien MME ne servent à rien puisque les identités SN id des deux réseaux ne sont pas les mêmes et l'authentification du réseau de service dans ce cas ne réussira pas lors de l'application d'une nouvelle procédure AKA. Le 3GPP autorise dans ce cas le transfert du contexte de sécurité actuel de l'ancien MME au nouveau MME pour que ce dernier puisse en profiter. C'est à l'opérateur du réseau dans ce cas de décider la politique de sécurité à adopter : si on transmet le contexte de sécurité ou on demande seulement l'IMSI de l'ancien MME.

3.5.4.3 Hiérarchie des clés

Après l'établissement de la clé K_{ASME} par la procédure EPS-AKA, toutes les clés nécessaires pour les différents mécanismes de sécurité seront générées à partir de cette clé. Ces clés forment une sorte d'arbre dont la hiérarchie est donnée par la figure 3.12. Cette hiérarchie inclut :

- la clé permanente K qui constitue la racine de l'arbre des clés,
- des clés intermédiaires qui servent à générer d'autres clés et qui sont : CK/IK, K_{ASME}, et K_{eNB},

- des clés feuilles K_{NASenc}, K_{NASint}, K_{RRCenc}, K_{RRCint} et K_{UPenc} qui servent comme clés des algorithmes de chiffrement ou d'intégrité pour assurer la protection de la signalisation NAS, AS et du plan usager (données).

Dans cette figure, les clés intermédiaires et les clés feuilles sont générées (dérivées) à partir des clés qui se trouvent juste en-dessus et avec lesquelles sont reliées par une flèche.

Une clé qui figure entre deux couches signifie qu'elle est générée dans le nœud de la couche adjacente supérieure et elle est envoyée au nœud de la couche inférieure. Par exemple la clé intermédiaire K_{eNB} est dérivée de la clé K_{ASME} et dans le MME (du côté réseau) et elle est envoyée à l'eNB dans le but de dériver d'autres clés feuilles pour la protection de l'accès radio. La flèche qui fait boucle sur la clé K_{eNB} représente une situation particulaire où une clé K_{eNB} d'une station de base eNB est calculée à partir d'une autre clé K_{eNB} d'une autre station eNB lors du passage d'une cellule à une autre (handover), sans revenir aux clés plus hautes dans la hiérarchie (K_{ASME}). Il s'agit de certaines situations de handover entre les eNB où le MME n'y est pas impliqué.

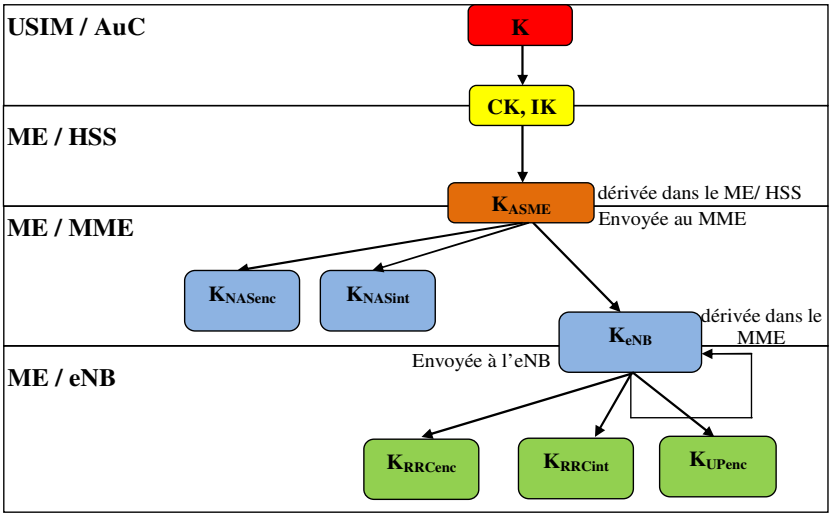

Figure 3.12. Hiérarchie des clés dans le système EPS

Contrairement à toutes les autres clés, la génération de CK et IK (à partir de la clé K), dans la carte USIM de l'utilisateur et à l'intérieur de l'AuC du côté réseau, se fait d'une manière qui n'est pas standardisée où elle compte sur les fonctions f3 et f4 qui ne sont pas prédéfinies. La raison est que les deux entités qui génèrent ces deux clés, c.à.d. l'USIM et l'AUC, sont contrôlées par le même opérateur. Tandis que pour le reste des clés, la situation est différente. Comme la génération se passe dans le ME du côté utilisateur, et dans le HSS, le MME, et l'eNB du côté réseau, donc il est obligatoire de standardiser les fonctions de sécurité qui produisent ces clés. On utilise une fonction de hachage à sens unique dans la dérivation des clés puisqu'il garantie qu'à partir des clés des couches inférieures, il est pratiquement impossible de calculer les clés des couches supérieures. Ce qui augmente bien le niveau de sécurité du réseau.

3.5.4.3.1 Dérivations et Objectifs des clés dans la hiérarchie

La figure 3.13 montre la génération (dérivation) des différentes clés, intermédiaires et feuilles, effectuée dans le ME du côté de l'utilisateur et dans les nœuds (HSS, MME et eNB) du côté

réseau. La fonction 'KDF' n'est que l'algorithme HMAC-SHA-256, et la fonction 'Trunc' désigne une fonction de troncature simple qui prend seulement les 128 bits les moins signifiants à partir d'une entrée de 256 bits, et jette le reste des bits. Les cinq clés feuilles utilisées sont des clés de 128 bits obtenues par des fonctions 'Trunc' et elles sont obtenues à partir les clés K_{eNB} et K_{ASME} qui ont une taille de 256 bits. Les dernières propositions (Release 10 et 11) de l'EPS, affirment que la sécurité fournie par les clés de 128 bits est considérée comme suffisante et la troncature est utilisée. Mais le jour où cette sécurité sera considérée comme insuffisante et on exige l'augmentation de la taille des clés feuilles, on élimine dans ce cas les fonctions Trunc et la taille devient 256 bits.

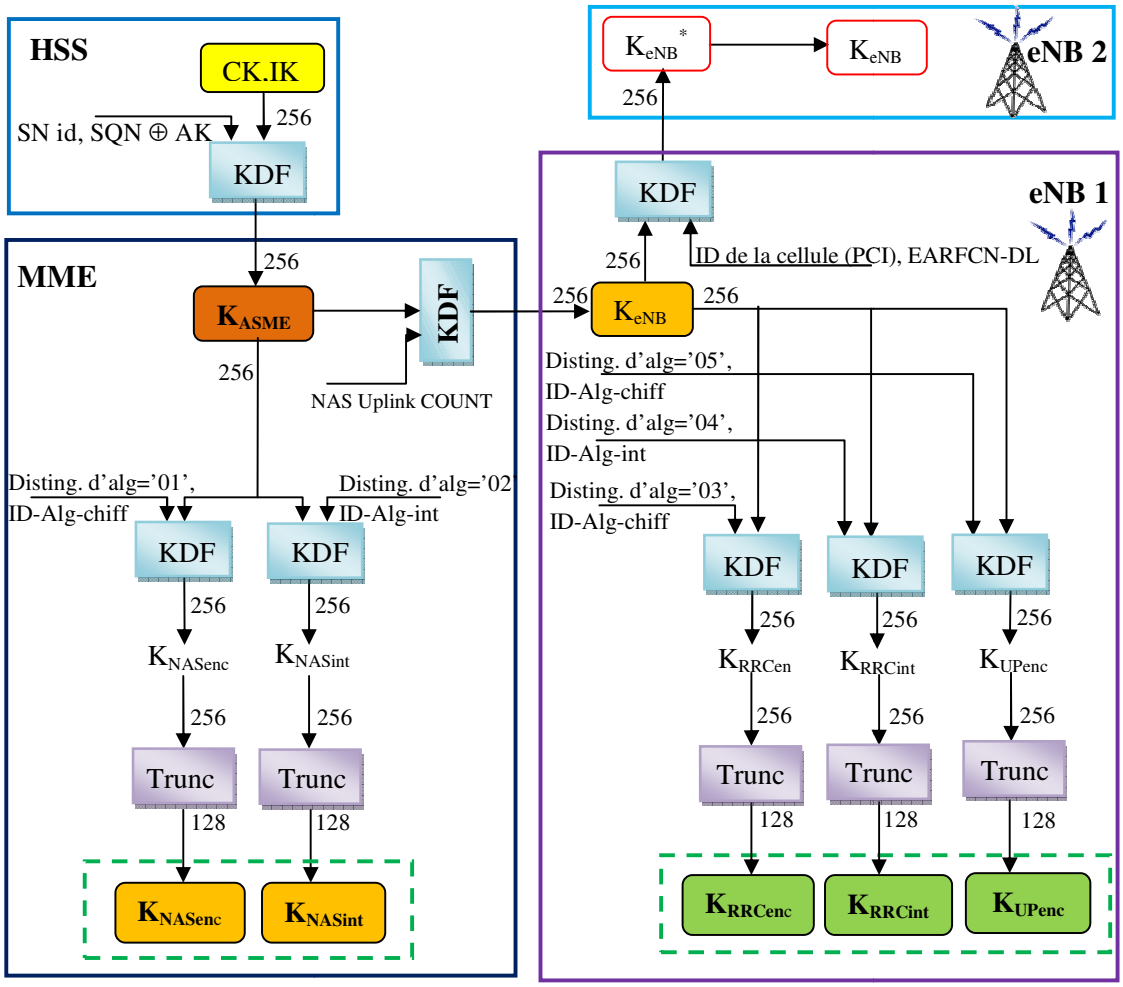

Figure 3.13. Chaîne de dérivation des clés en EPS du côté réseau.

Présentons maintenant le rôle de chacune des clés de la hiérarchie, ainsi que les paramètres d'entrée qui sont nécessaires à la dérivation de chacune d'elles :

- **K** : est la clé secrète permanente de 128 bits attribuée à l'usager, et connue seulement par l'USIM et par l'AuC. Elle ne quitte jamais ces deux entités et la hiérarchie des clés repose sur cette clé racine.

- **CK, IK** : sont des clés de 128 bits, dérivées de K, en utilisant les fonctions de sécurité f3 et f4 respectivement avec le paramètre d'entrée RAND. Ces clés sont transmises d'une part de l'AuC au HSS et d'autre part de l'USIM au ME.

- **K_{ASME}** : est la clé maîtresse obtenue par la procédure EPS-AKA et dérivée à partir de CK/IK. On utilise la fonction KDF pour générer cette clé ainsi que les deux paramètres suivants : l'identité du réseau de service SN id, utilisée pour relier la clé au réseau où elle est supposée être stockée et utilisée, et le paramètre (SQN $\oplus$ AK) retiré du jeton AUTN.

Les clés dérivées de K_{ASME} pour la protection de la signalisation NAS, échangées entre l'UE et le MME, et pour la génération d'autres clés sont :

- **K_{NASenc}** : est une clé utilisée pour chiffrer les messages de signalisation NAS. Elle est dérivée de K_{ASME} en utilisant la fonction KDF et deux paramètres d'entrée : le premier paramètre est appelé 'distingueur du type d'algorithme' et qui prend la valeur égale à '01', et le deuxième paramètre indique l'algorithme de chiffrement utilisé. Le paramètre 'distingueur du type d'algorithme' indique le rôle et la fonctionnalité de la clé, si elle sert pour le chiffrement ou pour l'intégrité, et pour quel type de données (NAS, RRC ou plan usager).

- **K_{NASint}** : est une clé utilisée pour protéger l'intégrité des messages de signalisation NAS. Elle est dérivée de K_{ASME} en utilisant la fonction KDF et deux paramètres d'entrée : le premier est appelé 'distingueur du type d'algorithme' et qui prend la valeur '02', et le deuxième est l'identifiant de l'algorithme d'intégrité utilisé.

- **K_{eNB}** : est générée à partir de K_{ASME} et de l'entrée supplémentaire 'NAS Uplink COUNT'. Ce dernier est un compteur de la liaison montante NAS qui sert à garantir que deux clés K_{eNB} , dérivées de la même K_{ASME} , ne peuvent jamais être les mêmes. Cette clé est la clé maîtresse locale dans un eNB qui sert à générer d'autres clés feuilles.

Les clés dérivées de K_{eNB} servent à protéger tout ce qui est transmis sur la partie radio entre l'UE et l'eNB, comme la signalisation RRC et le plan usager (données ou voix). Elles sont les suivantes :

- **K_{RRCenc}** : est la clé utilisée pour chiffrer les messages de signalisation RRC. Elle est générée en utilisant deux paramètres d'entrée : le premier est le 'distingueur du type d'algorithme' qui prend la valeur '03', et le second est l'identifiant de l'algorithme de chiffrement.

- **K_{RRCint}** : sert à protéger l'intégrité des messages de signalisation RRC. Elle est dérivée à partir de deux paramètres d'entrée : le premier est le 'distingueur du type d'algorithme' qui prend la valeur '04', et le second est l'identifiant de l'algorithme d'intégrité.

- **K_{UPenc}** : est utilisée pour le chiffrement des données du plan usager, et elle est dérivée de K_{eNB} et des deux paramètres suivants : le premier est le 'distingueur du type d'algorithme' qui prend la valeur '05', et le deuxième est l'identifiant de l'algorithme de chiffrement.

Les clés de chiffrement et d'intégrité NAS (K_{NASenc}, K_{NASint}) et AS (K_{RRCen}, K_{RRCint}, K_{UPenc}) sont liées aux algorithmes utilisés (ID-Alg). Chacune de ces clés feuilles change lorsque l'algorithme de chiffrement ou d'intégrité utilisé est modifié.

La dernière clé présentée dans la figure 3.13 est la K_{eNB}*. C'est une clé intermédiaire générée à partir d'une autre K_{eNB} lorsqu'il y a un handover. En effet, lors d'un handover et lorsque l'usager passe d'une cellule à une autre en changeant aussi la station de base eNB, l'eNB source détermine une nouvelle clé K_{eNB} , notée K_{eNB}*, calculée à partir de l'ancienne K_{eNB} source et d'autres paramètres. Ces derniers sont l'identifiant de la cellule (Cell Id) et la fréquence de la liaison descendante (EARFCN-DL) afin de relier la clé au contexte local. De cette façon, lorsque l'usager arrive à une nouvelle cellule et change de station de base eNB, il continue sa transmission en utilisant d'autres clés feuilles générées à partir de la nouvelle clé K_{eNB}* qui lui a été préparé avant de quitter son ancienne cellule.

Notons qu'il y a d'autres types de clés qui peuvent être utilisées au sein de l'EPS pour des fins diverses comme : l'interfonctionnement avec le système 3G et lors d'un handover de l'EPS au réseau 3G (on utilise dans ce cas des clés qu'on appelle CK' et IK'), et 2) dans le but d'augmenter la couverture LTE dans des zones où on implémente des nœuds de la dernière version de l'EPS [TS 33.401, 2012] comme : Relay Node (RN) et le DeNB (Donor eNB) (on utilise dans ce cas des clés qu'on appelle K_{UPint}).

3.5.4.3.2 Fonction de dérivation de clés KDF

Le 3GPP utilise la fonction de dérivation de clés KDF, spécifiée dans [TS 33.220, 2012] et [Holtmanns *et al.*, 2008], et qui est formée de la fonction de hachage approuvée SHA-256 [FIPS 180-4, 2012] utilisée dans le mode HMAC [FIPS 198-1, 2008]. Cette fonction KDF prend deux paramètres (*Kin*, *S*) à son entrée : la clé secrète *Kin* et le message *S*. Ce dernier contient tous les paramètres supplémentaires nécessaires à la dérivation. Le message *S* est définit comme une chaîne binaire ayant une structure bien définie et donnée par :

$$S = FC \parallel P0 \parallel L0 \parallel P1 \parallel L1 \parallel P2 \parallel L2 \parallel ... \parallel Pn \parallel Ln$$

Le premier champ FC (Function Code), formé d'un seul octet, a pour rôle de préciser l'utilisation prévue de la clé générée, c.à.d. si la clé dérivée sera utilisée pour : générer d'autres clés intermédiaires (comme le K_{ASME}), pour la dérivation des clés de chiffrement et d'intégrité (comme K_{NASenc}, K_{NASint}, etc), ou pour d'autres fins. Les valeurs allouées par le 3GPP au champ FC [TS 33.220, 2012] appartiennent à l'intervalle [0x10 ; 0x1F], dont les valeurs allant de 0x1C jusqu'au 0x1F sont réservées pour des futures utilisations. Les paramètres Pi, avec i=0... *n*, sont les paramètres d'entrée supplémentaires qui sont nécessaires pour la dérivation de la clé, et le paramètre Li, formé de deux octets, donne la longueur du paramètre Pi en octets. On peut avoir un seul paramètre P0, ou deux paramètres (P0 et P1) ou même plus. Le tableau 3.2 montre quelques exemples de génération de clés avec les valeurs de K_{in}, FC, Pi, et Li correspondant.

Clé dérivé= KDF (*Kin*, *S*), où KDF=HMAC-SHA-256		***S*= FC‖ P0‖ ‖L0‖ P1‖ L1‖….‖ P*n*‖ L*n***			**Longueur de la clé dérivée (bits)**
Clé(s) dérivée(s) à la sortie	Clé secrète d'entrée *Kin*	FC alloué [TS 33.401, 2012]	P0, P1, …., P*n*	L0, L1,..,L*n*	
K_{ASME}	CK‖IK	0x10	SNid, SQN⊕AK	0x0003, 0x0006	256
K_{eNB}	K_{ASME}	0x11	La valeur Count de la liaison montante NAS	0x0004	256
K_{eNB*}	K_{eNB}	0x13	PCI, EARFCN-DL	0x0002, 0x0002	256
K_{NASenc}, K_{NASint}, K_{RRCenc}, K_{RRCint}, K_{UPenc}, K_{UPint}	K_{ASME} ou K_{eNB}	0x15	Distingueur de type d'algorithme, Identifiant de l'algorithme	0x0001, 0x0001	128
CK', IK' dans handover	K_{ASME}	0x16	La valeur Count de la liaison descendante NAS	0x0004	128 chacun
…	…	….	…		…
CK', IK' dans la mobilité en veille	K_{ASME}	0x1B	La valeur Count de la liaison montante NAS	0x0004	128 chacun

Tableau 3.2 : Les paramètres d'entrée (*Kin*, *S*) des différentes KDF en EPS

Pour mieux clarifier ce que nous venons d'expliquer, prenons deux exemples de génération de clé. Comme premier exemple, considérons la dérivation de la clé K_{eNB} à partir de la clé maîtresse K_{ASME}. Dans ce cas, FC est définit comme égale à 0×11, et le seul paramètre d'entrée supplémentaire est la valeur COUNT de la liaison montante NAS. Donc P0 = la valeur de COUNT et comme sa longueur est de 4 octets, par suite L0 = 0×0004. Comme deuxième exemple, considérons la génération de la clé K_{NASint} à partir de la clé secrète K_{ASME}. Il a été décidé par le standard [TS 33.401, 2012] d'utiliser la valeur FC=0x15 non pas seulement pour l'exemple considéré mais aussi pour toutes les dérivations de clés qui mènent à une clé feuille dans l'hiérarchie des clés.

Dans notre cas, comme pour toutes les autres clés feuilles, il y a deux paramètres d'entrée supplémentaires nécessaires. Le premier paramètre est le 'distingueur du type d'algorithme' qui prend l'une des valeurs définis dans [TS 33.401, 2012] et données dans le tableau 3.3. Ce tableau contient toutes les valeurs allouées par le 3GPP pour les différents types d'algorithmes qu'on peut utiliser. Pour notre exemple la valeur de ce paramètre est P0=0x02, et comme sa longueur est de 4 bits seulement, donc étalé sur 1 octet, alors L0=0x0001. Le deuxième paramètre d'entrée de la KDF, représente l'identifiant de l'algorithme qui indique quel algorithme de chiffrement ou

d'intégrité nous utilisons. L'EPS utilise deux algorithmes standards pour le chiffrement et l'intégrité: SNOW-3G [TS 35.216, 2012] ou AES [FIPS 197, 2001]. Si on prend le SNOW-3G, la valeur de P1 sera égale à 0x01, et si on prend l'AES cette valeur sera égale à 0x02. Dans notre exemple et comme la valeur de P1 est étalée sur 1 octet, ce qui fait que L1=0x0001. La figure 3.14 montre le deuxième exemple considéré avec tous les paramètres d'entrée nécessaires.

Distingueur d'algorithme	Valeur
Alg-chiff-NAS	0x01
Alg-int-NAS	0x02
Alg-chiff-RRC	0x03
Alg-int-RRC	0x04
Alg-chiff-UP	0x05
Alg-int-UP	0x06

Tableau 3.3 : Distingueur des types d'algorithmes

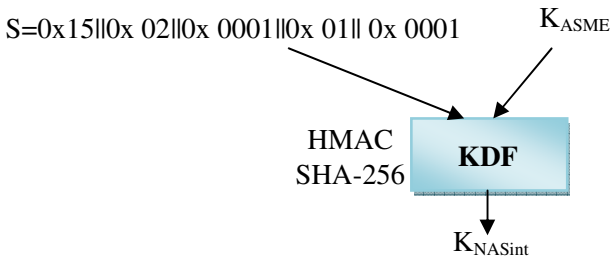

Figure 3.14. Dérivation de la clé K_{NASint} pour l'utiliser avec SNOW-3 G

3.5.4.4 Protection de la signalisation NAS, AS et des données usagers

L'EPS possède deux niveaux de sécurité pour la signalisation : le premier niveau, NAS entre l'UE et le MME, et le deuxième niveau AS entre l'UE et l'eNB. La protection de la signalisation consiste à chiffrer et à protéger l'intégrité des messages ainsi que la protection contre la répétition. Pour les données du plan usager, elles sont uniquement chiffrées entre l'UE et l'eNB (sans protection d'intégrité).

Le mécanisme de sécurité pour chacun des deux niveaux AS et NAS n'est pas appliqué qu'après l'application de la procédure d'établissement de la sécurité (AS et NAS). Ces procédures incluent chacune une commande de mode de sécurité qui déclenche et configure la sécurité dans chaque niveau.

3.5.4.4.1 Négociation des algorithmes de sécurité

Avant d'établir une communication sécurisée, l'UE et le réseau se mettent d'accord sur les algorithmes de sécurité à utiliser. Lors de l'enregistrement (l'attachement) de l'UE auprès du réseau, il envoie ses capacités de sécurité qui précisent les algorithmes de chiffrement et

d'intégrité implémentés sur son terminal. En release 8 [TS 33.401, 2011], il exige à l'UE et aux nœuds eNB et MME de supporter les algorithmes SNOW 3G et AES sur lesquels se basent les algorithmes de chiffrement et d'intégrité. Dans le dernier release (Release 11) [TS 33.401, 2012], un nouvel algorithme, conçu en Chine et appelé ZUC (Zu Chongzhi) [TR, 2011], est introduit.

Les nœuds MME et eNB choisissent les algorithmes à utiliser en se reposant sur les capacités de sécurité d'UE, et sur la liste configurée par l'opérateur des algorithmes de sécurité autorisée par ordre de priorité. En effet, l'UE envoie ses capacités de sécurité au réseau pendant la procédure d'attachement au réseau (dans le message *Attach Request*) et quand il envoie un message de mise à jour de localisation (*Tracking Area Update Request*).

Les capacités de sécurité envoyées par l'UE au réseau, lui sont renvoyées par le réseau dans un message de réponse NAS, protégée en intégrité. L'objectif de retourner les mêmes informations est la sécurité et la protection contre les attaques (*bidding down attack*), où l'attaquant modifie le message transportant les capacités de sécurité de l'UE vers le réseau. Si l'UE détecte une différence entre les capacités de sécurité, envoyées et reçues, il annule la procédure d'attachement. Le 3GPP recommande que ca serait encore mieux si la protection contre la modification (ou *bidding down attack*) se fait dans les deux sens.

Deux procédures de commande du mode de sécurité, une pour NAS et une autre pour l'AS, sont utilisées pour indiquer les algorithmes sélectionnés et pour activer le chiffrement et la protection de l'intégrité, ainsi que la protection de rejoue. Ceci, donne la possibilité d'utiliser en même temps des algorithmes différents pour la sécurité NAS et AS.

3.5.4.4.2 Protection de la signalisation NAS

3.5.4.4.2.1 Etablissement de la sécurité NAS

La figure 3.15 présente la procédure d'établissement de la sécurité NAS. Après la réception des capacités de sécurité de l'UE lors de la demande d'attachement, et après le déroulement de la procédure EPS-AKA, le MME choisit un algorithme de chiffrement et un algorithme d'intégrité pour la protection de la signalisation NAS. Les deux algorithmes sélectionnés doivent être disponibles et implémentés dans l'UE et dans le MME. Ensuite, le MME dérive les clés K_{NASenc} et K_{NASint}, et envoie le message 'Commande du mode de sécurité NAS' (appelé CMS NAS ou *NAS Security Mode Command*) à l'UE. Ce message contient : les capacités de sécurité d'UE que le MME vient de recevoir (pour des raisons de sécurité comme nous avons dit) ; les algorithmes NAS sélectionnés pour la protection des messages NAS ; le KSI_{ASME} pour identifier la clé K_{ASME} à utiliser dans la dérivation des clés (dans l'UE) ;

Ce message est protégé en intégrité via le champ NAS-MAC, mais il n'est pas chiffré puisque l'UE ne connaît pas encore avec quel algorithme il doit effectuer le déchiffrement. L'UE vérifie l'intégrité de ce message en se servant de la clé K_{NASint} (dérivée de K_{ASME} établie déjà et indiquée par le KSI_{ASME}), et en utilisant l'algorithme d'intégrité NAS indiqué dans le message qu'il vient de recevoir (CMS NAS). Puisque le MME connaît déjà les algorithmes et les clés qui ont été sélectionnés, il peut recevoir et déchiffrer des messages chiffrés venant de l'UE. Il peut donc commencer le déchiffrement de la signalisation NAS de la liaison montante juste après avoir envoyé le message 'CMS NAS'. S'il n'y a pas de problème dans le message CMS NAS reçu, l'UE envoie au MME le message 'mode de sécurité NAS établi' (*NAS Security Mode Complete*)

chiffré et protégé en intégrité. Dans ce message, l'UE peut envoyer au réseau son identifiant du terminal, l'IMEISV, chiffré. Ceci se fait si ce dernier a été demandé par le réseau, dans le message 'CMS NAS'.

Le MME peut commencer le chiffrement de la signalisation NAS émis sur la liaison descendante après réception et vérification du message 'Mode de sécurité établi'. Si le ME détecte que l'intégrité du message 'CMS NAS' n'est pas correcte, il envoie un message de rejet (*NAS Security Mode Reject*) [TS 24.301, 2011] protégé avec les clés NAS établis (avant l'émission du message CMS NAS). Or, lors du premier attachement il n'y a aucun contexte de sécurité NAS qui est encore établit, et dans ce cas s'il y a un message de rejet à envoyer, il sera émis sans aucune protection. Ceci présente un point faible que nous essayerons de résoudre dans le chapitre suivant.

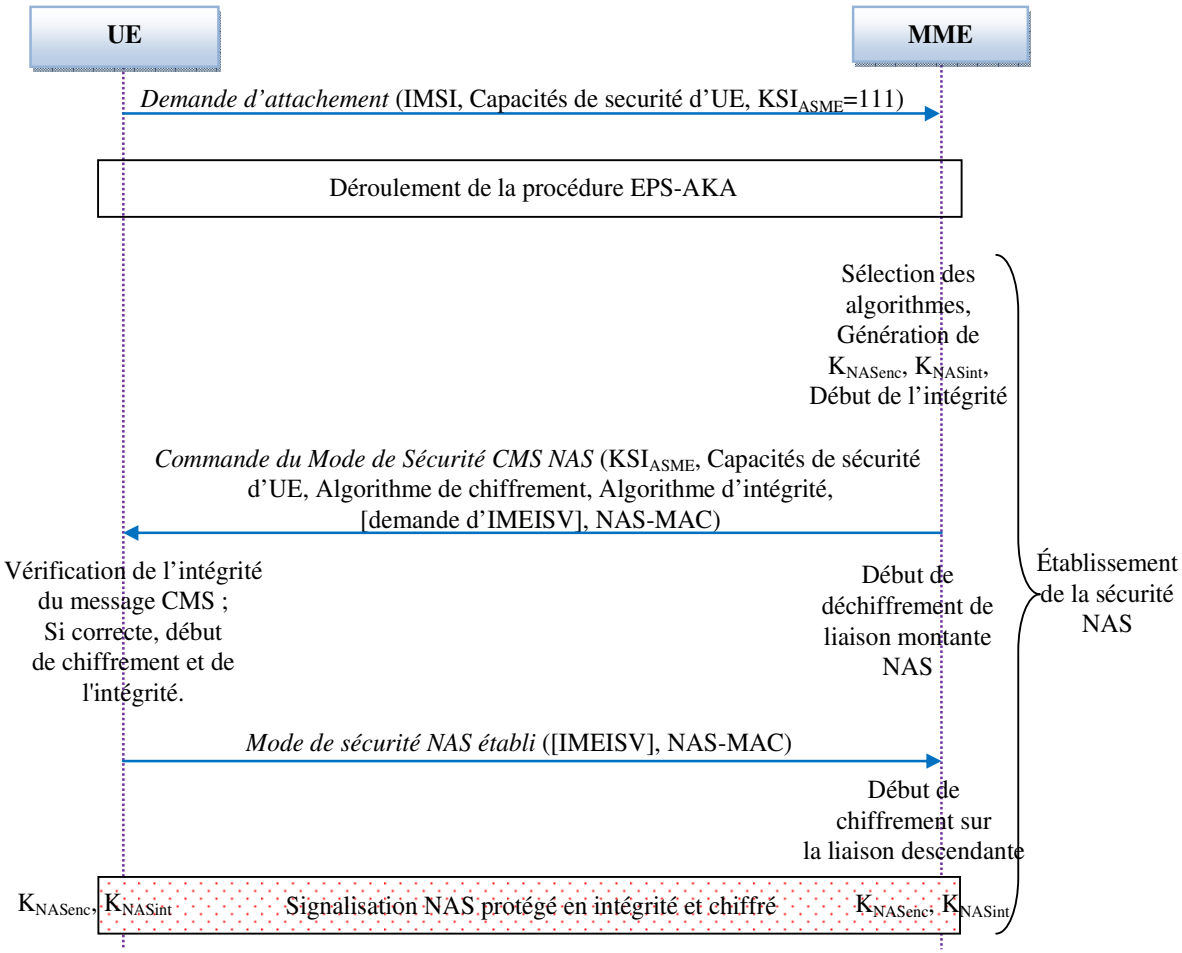

Figure 3.15. Procédure d'établissement de la sécurité NAS

3.5.4.4.2.2 Protection de l'intégrité des messages NAS

La protection de l'intégrité des messages NAS est effectuée par la couche NAS EMM, dans l'UE ou dans le MME, qui ajoute un code MAC à la fin des messages de signalisation NAS transmis. Ce code est calculé en utilisant la fonction de sécurité f9 (représentant l'algorithme d'intégrité

EIA) et la clé d'intégrité K_{NASint}. Plusieurs paramètres sont également appliqués à l'entrée de la fonction f9 afin d'assurer l'unicité du code MAC obtenu. La figure 3.16, montre la manière d'utilisation de cette fonction, et les entrées de l'algorithme d'intégrité sont les suivants :

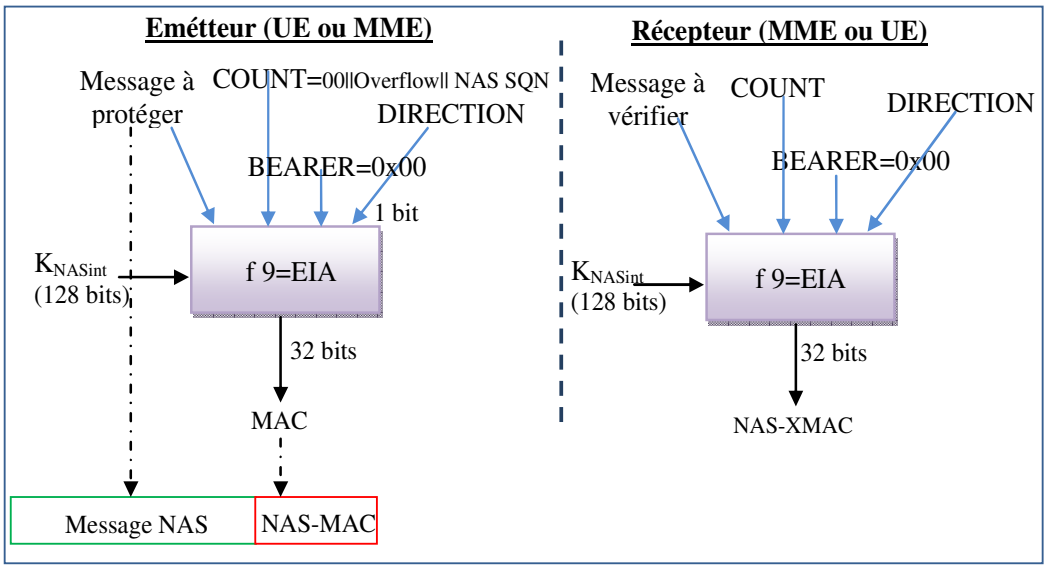

Figure 3.16. Mécanismes de protection et de vérification de l'intégrité des messages NAS

- K_{NASint} : La clé d'intégrité NAS de 128 bits ;
- Message : est le message NAS à protéger (comme par exemple le message CMS-NAS ou autre) ;
- COUNT : un compteur de séquence sur 32 bits, et qui est composé de trois parties concaténés de la manière suivante : COUNT= 0x00 || NAS OVERFLOW || NAS SQN. Avec NAS OVERFLOW est un champ de 16 bits qui s'incrémente à chaque fois que le compteur NAS SQN, de 8 bits, dépasse sa valeur maximale.
- BEARER : est l'identifiant, sur 5 bits, du bearer qui porte le message NAS transmis. Il prend une valeur constante égale à 0x00 puisque tous les messages NAS sont transportés sur un seul bearer.
- DIRECTION : est un seul bit qui indique le sens du message et qui prend la valeur '0' pour la voie montante et '1' pour la voie descendante. Ce champ est utilisé pour éviter l'utilisation des mêmes paramètres d'entrée pour les deux voies.

Le côté qui envoie le message, calcule le code MAC, NAS-MAC de 32 bits et l'ajoute à la fin du message NAS. Le récepteur effectue la même opération de calcul du code d'authentification à partir du message NAS reçu, et le résultat NAS-XMAC est ensuite comparé avec le NAS-MAC reçu. S'ils sont égaux alors le message NAS est accepté et s'ils sont différents le message NAS sera rejeté.

Cette protection protège aussi contre les attaques de répétition (replay protection) puisqu'il y a un compteur qui s'incrémente avec chaque message. Le récepteur accepte le message NAS si la valeur du champ COUNT n'a pas déjà été reçue et si elle est bien incrémentée.

Le standard exige l'implémentation de deux algorithmes d'intégrité EIA (EPS Integrity Algorithm) qui sont le SNOW 3G et l'AES, et il recommande d'implémenter un troisième algorithme qui est le ZUC [TS 35.221, 2012]. L'identifiant de l'algorithme d'intégrité, qui indique quel algorithme on utilise et qui consiste une entrée supplémentaire pour les fonctions de génération des clés KDF, est un champ de 4 bits et qui prend l'une des trois valeurs données par le tableau suivant.

Identifiant de l'algorithme d'intégrité (ID-Alg-int)	Algorithme d'intégrité
'0001': EIA1	SNOW 3G
'0010': EIA2	AES
'0011': EIA3	ZUC

Tableau 3.4. Les algorithmes d'intégrité EPS (EIA)

3.5.4.4.2.3 Chiffrement des messages NAS

Le chiffrement des messages NAS, présenté dans la figure 3.17, est effectué dans la couche NAS EMM avec la fonction de chiffrement f8 comme algorithme EEA (EPS Encryption Algorithm). Ce dernier utilise les mêmes paramètres d'entrée (COUNT, BEARER, DIRECTION) que l'algorithme d'intégrité (EIA), à l'exception de la clé, qui est K_{NASenc} pour le chiffrement, et le paramètre supplémentaire 'LENGTH'. Ce dernier est un champ de 16 bits qui sert à indiquer la longueur du bloc de chiffrement.

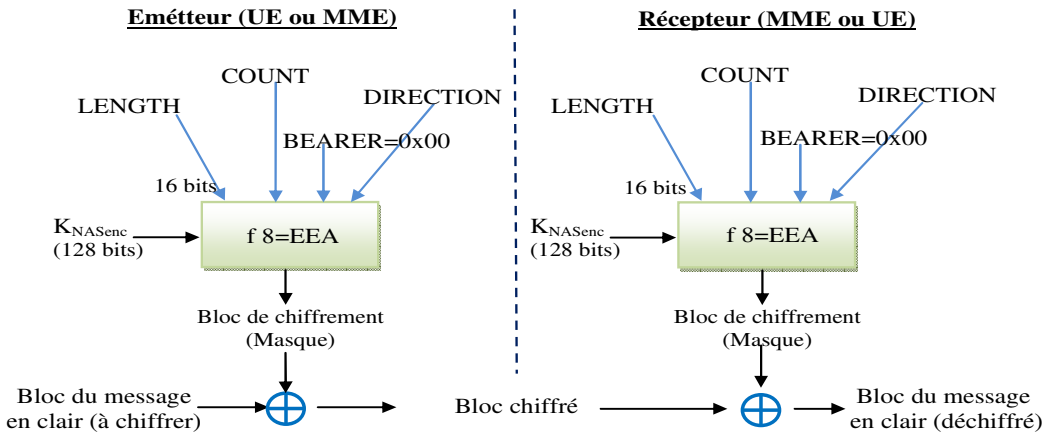

Figure 3.17. Mécanismes de chiffrement et déchiffrement des messages NAS

L'opération de chiffrement (par flux) consiste à additionner bit par bit, par ou exclusive, un bloc du message NAS en clair, à un bloc de chiffrement (Keystream Block) généré par f8 et de même longueur (LENGTH). De manière similaire, le déchiffrement est une opération symétrique. Il est effectué avec l'opération ou exclusive entre le bloc reçu et le même bloc de chiffrement (Keystream Block) généré à partir de la même f8 et le même ensemble de paramètres.

En chiffrement, on utilise les mêmes algorithmes utilisés en intégrité. L'identifiant de l'algorithme de chiffrement est aussi un champ de 4 bits qui prennent l'une des valeurs définies dans le tableau suivant.

Identifiant de l'algorithme de chiffrement (ID-Alg-chiff)	**Algorithme de chiffrement**
'0000': EEA0	Pas de chiffrement
'0001': EEA1	SNOW 3G
'0010': EEA2	AES
'0011': EEA3	ZUC

Tableau 3.5. Les algorithmes de chiffrement EPS (EEA)

L'UE, l'eNB et le MME doivent implémenter les deux algorithmes SNOW 3G et l'AES, et le dernier algorithme ZUC est facultatif.

3.5.4.4.3 Protection de la signalisation AS et des données usagers

3.5.4.4.3.1 Etablissement de la sécurité AS

Après l'établissement de la sécurité NAS, le MME calcule la clé K_{eNB} , et l'envoie à l'eNB avec les capacités de sécurité d'UE comme indiqué dans la figure 3.18. L'eNB choisit un algorithme de chiffrement et un algorithme d'intégrité parmi les algorithmes disponibles dans l'UE. Les identifiants des algorithmes sélectionnés contribuent, en tant que paramètre d'entrée, à la dérivation des clés AS (K_{RRCenc}, K_{RRCint} et K_{UPenc}). Ensuite l'eNB envoie le message 'Commande du mode de sécurité AS', CMS AS, pour indiquer à l'UE les algorithmes AS sélectionnés et pour annoncer le déclenchement de la sécurité. Ce message est protégé en intégrité via le champ MAC-I, généré à l'aide de K_{RRCint}. De son côté, l'UE dérive les 3 clés du niveau AS et, en utilisant l'algorithme d'intégrité indiqué dans le message CMS AS reçu, il vérifie avec la clé K_{RRCint} si le code MAC-I est correcte ou non. Si oui, l'UE répond avec le message 'Mode de sécurité AS établi', protégé en intégrité et contre la répétition, et il se prépare à recevoir des messages RRC et des données usagers chiffrés sur la liaison descendante. Sinon, l'ME répond avec un message d'échec [TS 36.331, 2011]. Notons que l'algorithme de chiffrement AS (EEA) sélectionné par l'eNB est utilisé à la fois pour le chiffrement du trafic RRC et des données usagers. Et notons aussi que l'UE ne commence pas le chiffrement des messages RRC et des données usagers sur la liaison montante, avant qu'il envoie le message 'Mode de sécurité AS établi' à l'eNB.

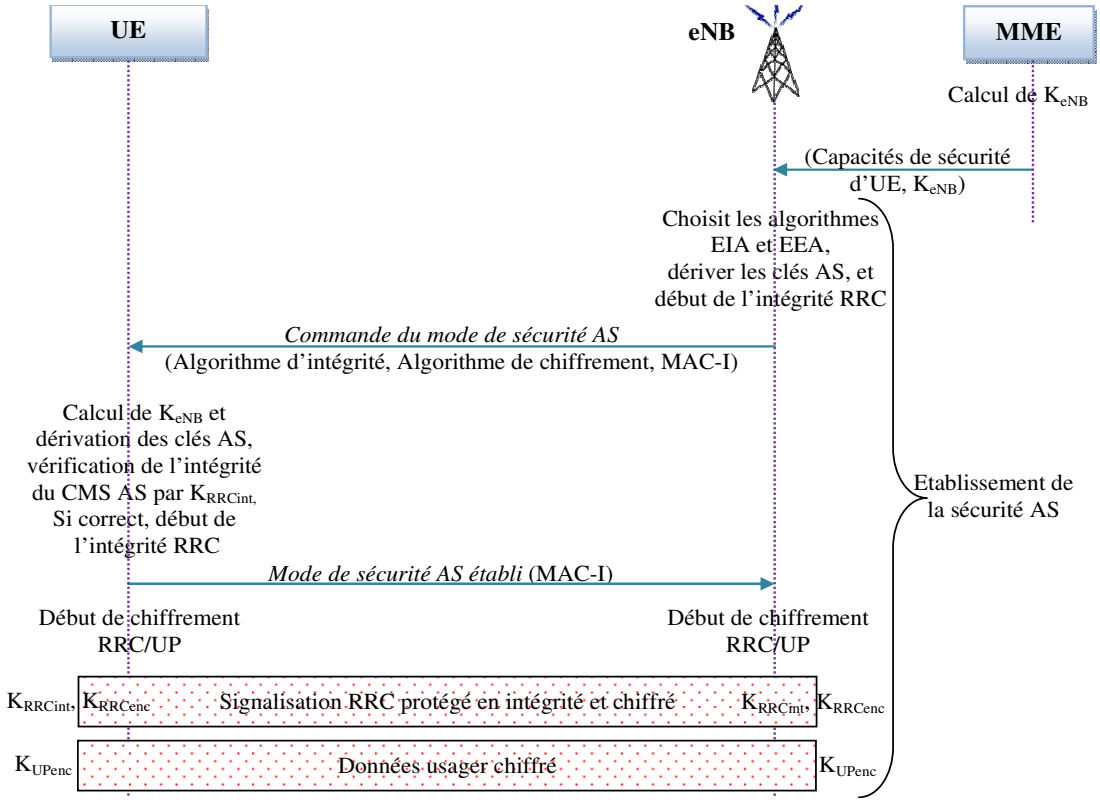

Figure 3.18. Établissement de la sécurité AS

3.5.4.4.3.2 Protection de l'intégrité des messages RRC

La signalisation AS est envoyée selon le protocole RRC [TS 36.331, 2011], et les messages RRC sont protégés en intégrité dans la couche PDCP. Le RRC et le PDCP forment deux couches adjacentes, l'une au dessous de l'autre suivant l'architecture du modèle OSI (voir figure 3.20). Notons que le RLC (Radio Link Control) est chargé de la segmentation/concaténation des paquets de données des couches supérieures, de la retransmission des paquets perdus [Tien Thinh, 2010], afin d'assurer la fiabilité du transport de données entre les entités.

Le même mécanisme de protection et de vérification d'intégrité utilisé pour les messages NAS est appliqué sur les messages RRC avec presque les mêmes paramètres d'entrée (voir figure 3.19) mais avec un algorithme d'intégrité (EIA) qui n'est pas nécessairement le même. Les quelques petites différences par rapport à la protection d'intégrité NAS sont :

1- la clé d'intégrité est la K_{RRCint} au lieu de K_{NASint},

2- la valeur de BEARER n'est pas constante et peut prendre plusieurs valeurs [TS 36.323, 2012] puisque le RRC a trois porteuses allouées pour la signalisation SRB (Signalling Radio Bearer).

3- le paramètre COUNT, formé de 32 bits, correspond au numéro de séquence introduit par le protocole PDCP, PDCP COUNT. Ce dernier est composé de deux champs concaténés

comme suivant : PDCP COUNT= HFN Overflow ||PDCP SQN, avec le PDCP SQN (12 bits) est un compteur envoyé dans les messages émis et le HFN (Hyper Frame Number) est un autre compteur qui s'incrémente à chaque fois le SQN dépasse sa valeur limite. Ce paramètre assure la protection contre la répétition.

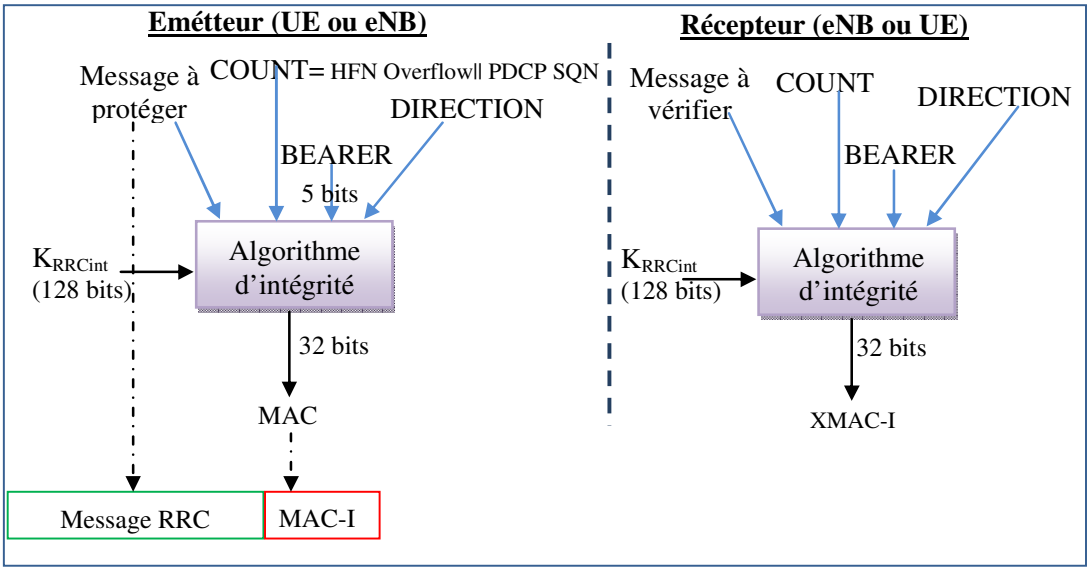

Figure 3.19. Protection et vérification de l'intégrité des messages RRC

3.5.4.4.3.3 Chiffrement des messages RRC et des données usagers

Les données du plan usager et les messages de signalisation RRC sont transportés selon le protocole PDCP [TS 36.323, 2012]. La sécurité est implémentée dans la couche PDCP et non pas dans la couche RRC ni dans le plan usager au-dessus de PDCP. Ceci diffère de la sécurité des messages NAS, qui fait partie du protocole NAS lui-même.

Puisque les messages NAS sont aussi transportés par le protocole PDCP entre l'UE et l'eNB. Ces messages doivent se protéger deux fois : la première au niveau NAS et la deuxième fois au niveau AS, et ceci après l'activation de ces deux niveaux de sécurité.

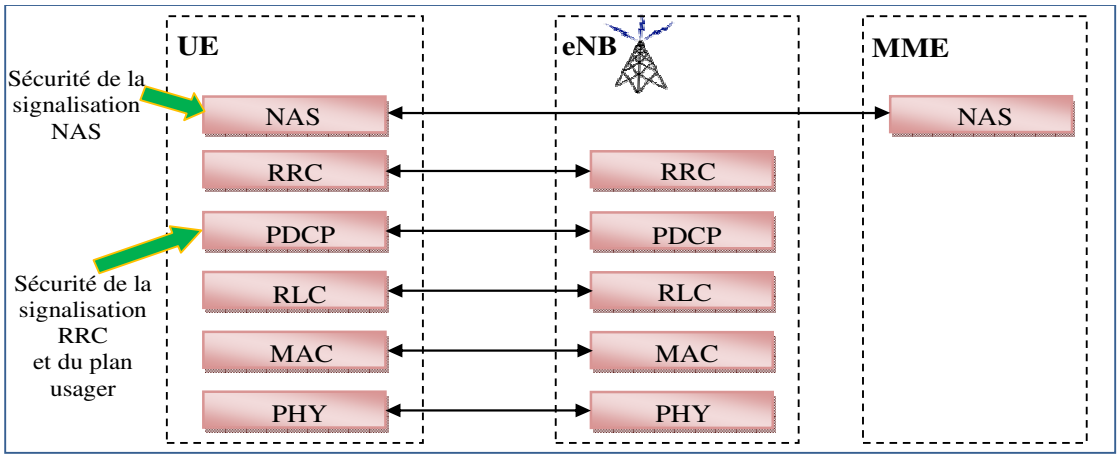

Figure 3.20. Position des services de la sécurité dans la pile des protocoles EPS

Le chiffrement des messages RRC et des données du plan usager utilisent le même mécanisme de chiffrement que les messages NAS (présenté dans la Figure 3.17), mais avec la clé K_{RRCenc} ou K_{UPenc}, au lieu de K_{NASenc} et avec les mêmes paramètres d'entrée. Les seules différences sont les valeurs des paramètres d'entrée : BEARER qui n'est pas constant, et COUNT qui est celle de PDCP COUNT comme définies dans le paragraphe précédent.

Notons qu'il est important de résumer les principales différences de la sécurité entre UMTS et EPS. Pour cela nous présentons cette comparaison dans l'annexe B2.

3.6 Conclusion

Dans ce chapitre nous avons présenté l'architecture des deux réseaux de téléphonie mobile, de troisième génération UMTS et de quatrième génération EPS (LTE/EPC). Nous avons présenté les mécanismes de sécurité 3G qui constituent une base solide pour la sécurité EPS. La sécurité 4G peut être vue comme une variante améliorée de la sécurité 3G, où on conserve ses avantages et ses éléments robustes et on traite ses faiblesses.

Nous avons étudié l'architecture de la sécurité en EPS qui a introduit de nombreuses nouvelles extensions et améliorations par rapport au 3G. Nous avons vu que la sécurité EPS est composée de cinq domaines d'application de la sécurité : l'accès au réseau, le domaine réseau, le domaine utilisateur, le domaine application, et la visibilité et la configuration de la sécurité par l'utilisateur. Nous avons montré que ces domaines collaborent ensemble pour offrir la sécurité dans tout le réseau EPS. Nous avons vu également comment les fonctions de sécurité offerte par cette architecture ont répondu à la majorité des exigences de la sécurité EPS, et quelles sont les exigences qui n'ont pas été satisfaites.

Comme le maillon le plus faible de n'importe quel réseau de télécommunication sans fil est la partie de l'accès au réseau, nous l'avons étudié en détail en expliquant les procédures et les mécanismes utilisés en EPS pour assurer un accès sécurisé au réseau. Ceci commence par les fonctions d'identification des abonnés et des terminaux, ensuite ca continue par la procédure d'authentification et d'établissement d'une clé maîtresse. Ensuite, vient l'utilisation d'une nouvelle hiérarchie pour la dérivation des clés EPS, et enfin ca se termine par l'établissement de la protection de la signalisation NAS, AS et des données usagers.

4. Analyse et amélioration de la sécurité de l'EPS

4. Analyse et amélioration de la sécurité de l'EPS

4.1 Introduction

Dans ce chapitre, nous analysons les faiblesses de la sécurité EPS et surtout les vulnérabilités du protocole EPS-AKA de 3GPP qui constitue la pierre angulaire de toute l'architecture de sécurité et qui permet l'accès des abonnés au réseau. Nous montrons, les faiblesses identifiées dans la littérature spécialisée, et nous avons nous-mêmes identifié de nouvelles faiblesses en ce protocole qui peuvent conduire à des attaques malveillantes contre les abonnés et le réseau. Les risques résultant de ces attaques peuvent : empêcher les utilisateurs d'accéder le réseau, provoquer l'usurpation des identités, provoquer le blocage des services de l'opérateur, etc.

Nous avons étudié et analysé cinq protocoles existants dans la littérature qui essayent d'améliorer la sécurité en EPS. Les deux premiers protocoles (EMSUCU, et Enhanced EMSUCU) [Al-Saraireh *et al.*, 2006] [Caragata *et al.*, 2011-a] traitent le problème de la transmission de l'IMSI en claire. Ils protègent tous les deux, cette identité par le chiffrement symétrique afin d'éviter les risques d'identification des utilisateurs. Le troisième protocole étudié est le standard de 3GPP, l'EPS-AKA. Ses différentes faiblesses ont été identifiées ainsi que les différentes attaques qui peuvent être montées sur ce protocole. Plusieurs protocoles ont été proposés pour le remplacer et résoudre ces vulnérabilités [Xiehua *et al.*, 2011] [He *et al.*, 2008] [Bou Abdo *et al.*, 2012-a] [Cho *et al.*, 2012]. L'analyse des deux meilleurs protocoles montre également des petites vulnérabilités. Notre objectif est de proposer un nouveau protocole AKA générique pour être utilisé comme une alternative de protocole 3GPP EPS-AKA.

Dans cette optique, nous avons proposé notre propre protocole qui apporte un jeu d'améliorations afin de résoudre tous les problèmes et les faiblesses identifiés dans l'EPS et les autres protocoles étudiés. Ce protocole, que nous avons appelé FP-AKA (Full Protection-AKA) a été évalué et a été comparé avec les autres protocoles considérés, selon quatre paramètres. Ces paramètres sont : le risque, le coût, le taux des données ajoutées sur les messages de signalisation, et le délai. Nous allons montrer que FP-AKA a le meilleur résultat selon les deux premiers paramètres et un très bon résultat dans les deux paramètres restants.

Il est important avant de commencer le chapitre, de rappeler les menaces qui n'ont pas été satisfaites par les fonctions de sécurité standardisées de l'EPS ou qui ont été traités partiellement sont les suivantes :

A) Menaces contre l'identité de l'utilisateur (*IMSI catching attack*) : l'utilisation des identités temporaires GUTI assure la protection contre cette menace et contre les attaques passives. Par contre les attaques actives peuvent s'effectuer facilement pour voler l'IMSI.

B) Menaces de suivi d'UE (*Threats of UE tracking*): satisfait partiellement par la fonction de confidentialité de l'identité de l'utilisateur et du terminal, grâce au chiffrement des GUTI. Ceci empêche l'attaquant d'observer les GUTI successives qui appartiennent à un même utilisateur. Mais ceci n'empêche pas le vol actif de l'IMSI.

C) Menaces contre la manipulation des données de signalisation (*Threats against manipulation of control plane data*): Il est vrai qu'on assure l'intégrité des messages de signalisation après le

lancement de la procédure EPS-AKA et après l'établissement des clés NAS et AS, mais durant ou avant la procédure AKA, les messages échangés ne sont pas bien protégés.

D) Menaces liées à un déni de service (*Threats related to denial of service*) : L'EPS souffre de ce type d'attaques qui peuvent être montées facilement. Comme exemples, on peut lancer une attaque distribuée à partir de plusieurs UE vers certaines parties du réseau, ou on peut effectuer des attaques DoS contre un UE. Si le réseau ne peut pas authentifier les messages qu'ils reçoivent, un attaquant peut envoyer un message modifié ou de déconnexion pour un autre mobile. Ceci a pour effet de couper la connexion entre le mobile en question et le réseau.

Dans la première partie de ce chapitre nous allons aborder les deux premières menaces, et dans la deuxième partie nous allons voir comment les deux autres menaces (C et D) affectent la procédure EPS-AKA. Nous expliquerons aussi quelles sont les solutions adéquates que nous proposons afin de les éviter.

4.2. Transmission de l'IMSI en claire

Un des premiers points faibles identifié de la sécurité de LTE, et hérité de l'UMTS, est l'envoie de l'identité permanente IMSI en claire [TR 33.821, 2009]. La transmission de l'IMSI se fait lors de la première connexion RRC établie lorsque le mobile est mis sous tension ou lors d'une demande par le réseau de service (VLR/SGSN en 3G ou MME en 4G). Le réseau de service demande l'identité de l'UE dans certains cas particuliers et sans l'obtention de l'IMSI, l'utilisateur serait définitivement exclu du système. La demande d'identité IMSI auprès du réseau de service se fait dans les deux cas suivants :

a) Passage de l'utilisateur d'un réseau de service à un autre lors d'une mise à jour de localisation, et indisponibilité de liaison entre ces réseaux, à cause d'un problème qui a touché par exemple le réseau d'où vient l'abonné ;

b) Perte de l'identité temporaire TMSI (3G) ou GUTI (4G), suite à un problème ou une panne dans le réseau de service, et le mobile voudrait faire un appel ou une mise à jour de localisation à l'intérieur du même réseau de service avec son TMSI/GUTI.

La transmission en claire permet à un attaquant qui écoute le canal radio de connaître l'identité de l'abonné et par suite d'ouvrir la voie à différents types d'attaques. L'attaque passive (écoute indiscrète) est la première méthode de l'interception de l'IMSI surtout dans des zones particulières bien connues comme l'aéroport où la majorité des nouveaux arrivants allument leurs mobiles. La deuxième méthode est l'attaque active qui a recours à certaines actions afin de découvrir l'identité IMSI [Arapinis *et al.*, 2012]. Dans une attaque active, l'obtention de l'IMSI peut se faire d'une manière très simple comme par exemple (voir figure 4.1) : l'attaquant utilise un appareil portatif de haute technologie appelé 'IMSI catcher' (capteur d'IMSI) [Strobel, 2007], qui joue le rôle d'une fausse station de base Node B ou eNB et prétend d'être un réseau de service légitime. Dans ce cas, lorsque l'utilisateur allume son téléphone mobile ou même dans certains cas lorsqu'il est en mode de veille, il se connecte à cette station de base qui lui demande (en lui transmettant le message *Identity Request message*) de s'authentifier avec son identité permanente IMSI. Ensuite l'UE doit obligatoirement, et pour ne pas se faire exclure du réseau, répondre en envoyant son identité IMSI.

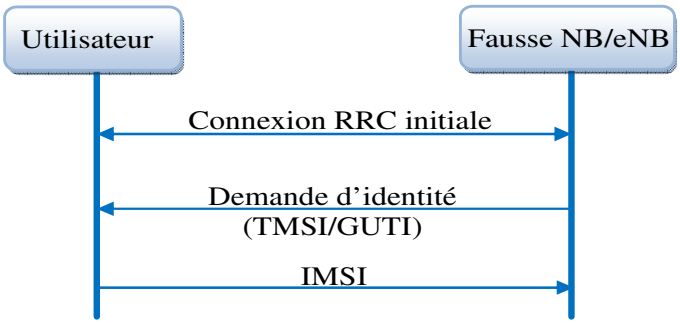

Figure 4.1. Obtention de l'IMSI par une attaque active

L'envoi du message *Identity Request message* par le 'capteur d'IMSI' consiste à tromper les téléphones mobiles en les laissant croire qu'ils s'inscrivent auprès d'un réseau de service légitime. La connaissance de l'IMSI par le pirate peut être utile pour des fins criminelles (selon le rapport alarmant de l'Europe en 2012), ou pour suivre les mouvements d'une personne et avoir sa localisation en permanence. Le capteur IMSI le plus sophistiqué est le "Stingray" qui coûte entre 60.000 et 175.000 dollars [Walker, 2013].

Notons que même si l'authentification qui s'effectue est mutuelle, ceci ne change rien de ce qui a été expliqué puisque l'utilisateur doit s'identifier avant qu'il authentifie le réseau.

Les solutions existantes à ce problème consistent à chiffrer l'IMSI lors de sa transmission sur la voie radio. Ces solutions sont basées, soit sur un chiffrement à clés symétriques comme [Al-Saraireh *et al.*, 2006] et [Caragata *et al.*, 2011-a], soit sur un chiffrement à clés publiques comme [Xiehua *et al.*, 2011] et autres.

Nous allons traiter dans cette partie les solutions basées sur le chiffrement symétrique, et nous allons voir dans la suite les solutions asymétriques lorsque nous analyserons les protocoles AKA proposés. En effet, les solutions asymétriques sont proposées en tant qu'une partie du protocole AKA, tandis que les solutions symétriques que nous allons traiter tout de suite proposent des remèdes hors le protocole AKA.

4.2.1 Solution d'Al-Saraireh - EMSUCU

La solution EMSUCU (Enhancement Mobile Security and User Confidentiality) proposée par Al-Saraireh en 2006 [Al-Saraireh *et al.*, 2006] pour l'UMTS s'applique aussi pour l'EPS. Elle consiste à chiffrer l'IMSI d'une manière symétrique avec une nouvelle fonction de sécurité, appelée f10 [TS 33.103, 2001], en suivant les étapes présentées dans la figure 4.2.

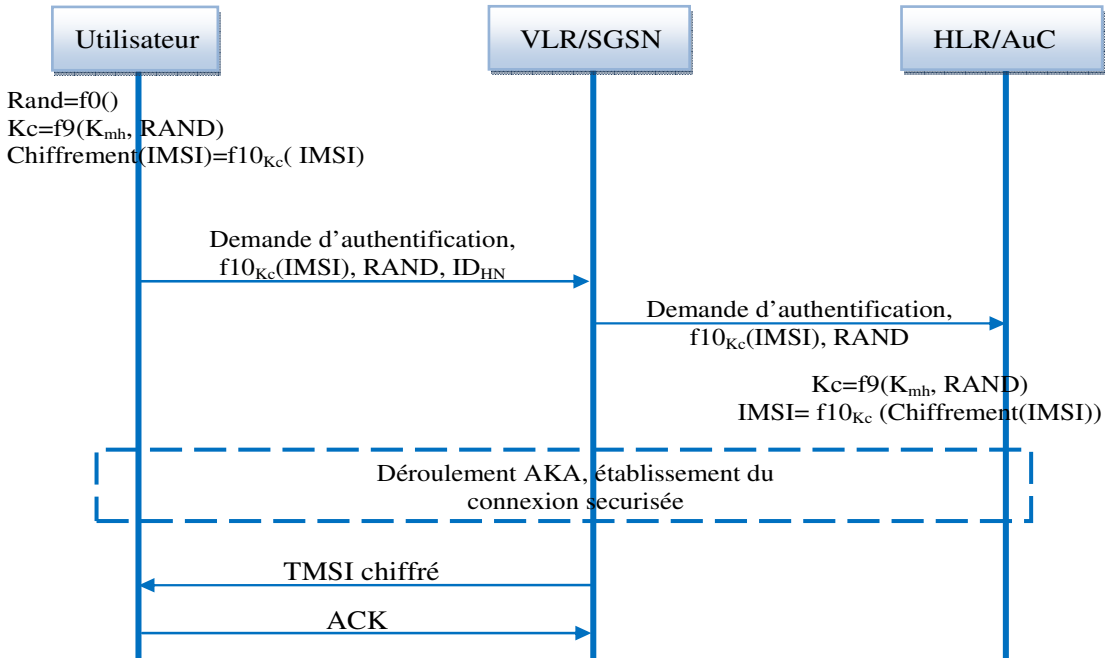

Figure 4.2. Solution proposée par Al-Saraireh (EMSUCU) pour la sécurité d'IMSI

Cette solution EMSUCU, suppose que l'USIM utilise la fonction f0 pour générer une valeur aléatoire RAND, et elle suppose aussi qu'une fonction de hachage f9 est utilisée pour la génération d'une clé Kc qui servira à chiffrer l'IMSI. La fonction f9 a deux paramètres d'entrée : une clé secrète K_{mh} et la valeur RAND. La clé K_{mh} est une clé unique partagée entre le HLR avec tous les clients. Lorsque le HLR reçoit la demande d'authentification, il utilise la valeur RAND reçue et la clé K_{mh} qu'il possède afin de calculer la valeur de la clé Kc. Cette valeur est ensuite utilisée pour le déchiffrement de l'IMSI et pour l'identification de l'UE. Le message envoyé par le mobile au réseau de service (VLR/SGSN) contient l'identifiant de son réseau d'origine, ID_{HN}, qui permet au VLR/SGSN de diriger le message vers le bon réseau d'origine.

4.2.2 Solution de Caragata - EEMSUCU

Différentes vulnérabilités du protocole EMSUCU ont été identifiées par Caragata [Caragata et *al.*, 2011-a]. Ce dernier a ajouté quelques améliorations sur l'algorithme d'Al-Saraireh [Saraireh et *al.*, 2006] et il a proposé un nouveau algorithme appelé Enhanced EMSUCU (ou EEMSUCU). Ce dernier n'est qu'une variante améliorée [Caragata et *al.*, 2011-a] de la solution d'Al Saraireh. Il est présenté par la figure 4.3 et les améliorations proposées sur l'EMSUCU sont les suivantes :

1) Protection de l'intégrité des messages de contrôle : Comme nous venons de voir dans l'algorithme EMSUCU, le message de la demande d'authentification transmis par l'UE au réseau de service, n'a aucune protection d'intégrité. Ce message est par suite vulnérable à des attaques actives comme l'attaque de Man-In-The-Midle MITM (l'homme au milieu). La première amélioration consiste à assurer l'intégrité de ce message en utilisant la clé Kc où un code d'intégrité MAC est calculé pour le message

{$f10_{Kc}$(IMSI), Rand} contenant l'IMSI chiffré et la valeur Rand. Ceci se fait en utilisant une fonction de hachage f12, choisie de manière qu'elle soit rapide et robuste.

2) La deuxième amélioration consiste à remplacer la clé K_{mh}, qui est partagée entre le HLR avec tous les utilisateurs, par la clé K (clé partagée entre chaque utilisateur et son réseau d'origine) afin de générer la clé Kc (voir figure 4.3). En effet, si un attaquant arrive à détecter la clé K_{mh}, il peut facilement déchiffrer tous les IMSI de tous les utilisateurs du HLR en cause, puisque le RAND est transmis en clair. C'est la raison pour laquelle K est utilisée à la place de K_{mh}.

3) La troisième amélioration proposée par Caragata, consiste à augmenter la taille de la clé de chiffrement Kc et rendre sa taille égale à 128 bits [Ecrypt II, 2011] au lieu de 64 bits. Pour cela on utilise une nouvelle fonction de hachage f11 au lieu de f9.

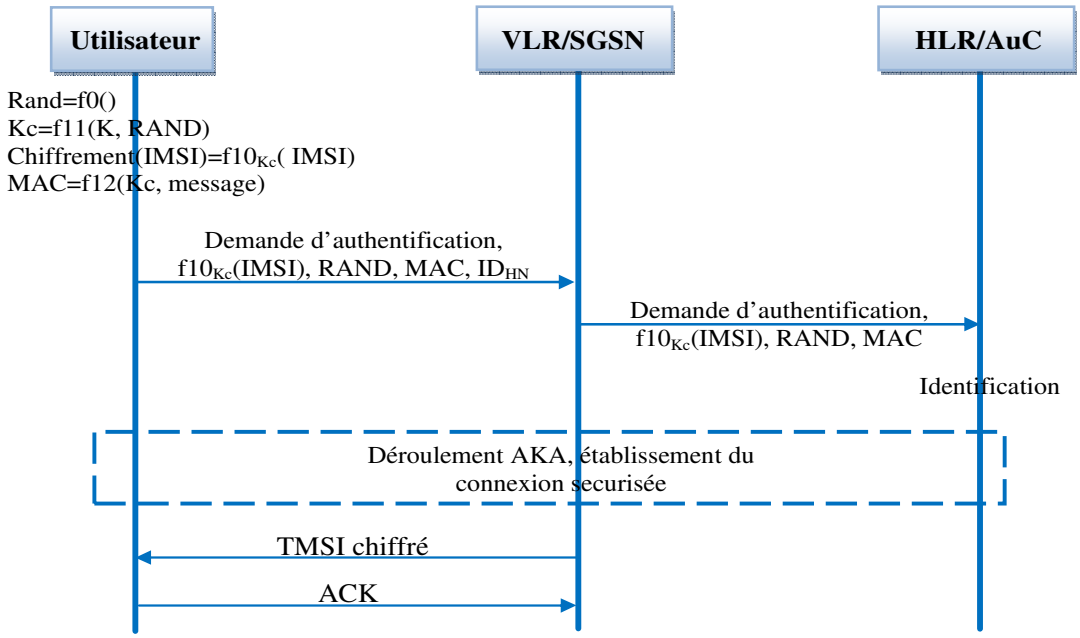

Figure 4.3. Solution de Caragata pour la sécurité d'IMSI (Enhancement on EMSUCU)

La procédure d'identification de l'abonné effectuée par le HLR/AuC, consiste à chercher dans tous les mobiles qui sont éteints afin d'identifier le mobile en trouvant la bonne clé Ki qui donne le bon code d'intégrité MAC. La recherche dans les mobiles éteints uniquement, signifie que l'auteur suppose que le mobile était obligatoirement éteint et il vient de se mettre sous tension pour établir la première connexion RRC. L'organigramme donné par la figure 4.4 résume cette procédure d'identification de l'abonné.

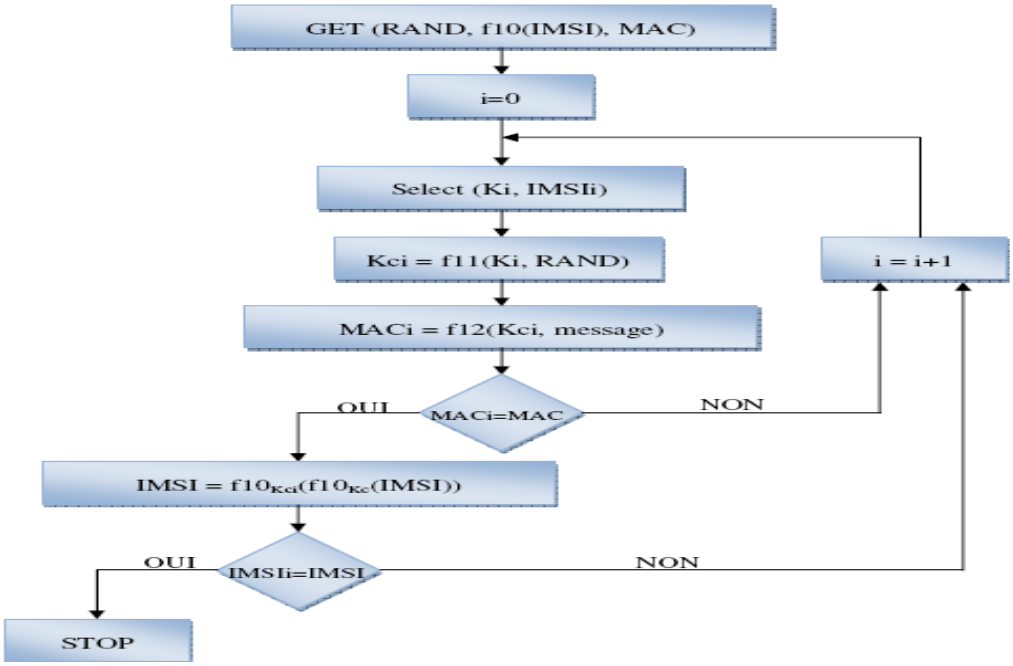

Figure 4.4. Procédure d'identification des utilisateurs

Pour chaque mobile éteint (ou enregistrement des mobiles éteints), et dans le but d'identifier l'utilisateur (et trouver l'IMSI correspondant) le HLR/AuC effectue les 3 opérations suivantes :

1- Génération, par la fonction f11, de la clé de chiffrement Kci
2- Calcul, par la fonction f12, du MACi correspondant
3- Comparaison du MACi avec le MAC reçu.

Si le MACi calculé et le MAC reçu ne sont pas égaux, il passe à l'abonné (à l'enregistrement) suivant. S'ils sont égaux, et à l'aide de la clé Kci et de la fonction f10, il découvre l'identité de l'abonné en déchiffrant l'IMSI reçu, alors il s'arrête et l'abonné est identifié. Le HLR/AUC effectue les 3 opérations citées jusqu'à que le code MACi soit égale au MAC reçu. Donc, pour chaque abonné et pour chaque test nous devons effectuer deux opérations consommables en utilisant les deux fonctions de hachage f11 et f12 afin de faire le test nécessaire.

4.2.3 Problématique d'EEMSUCU : vulnérabilités et attaques possibles

Dans cette section nous analysons l'algorithme EEMSUCU, en expliquant ses vulnérabilités, ainsi que les attaques possibles qui peuvent le casser. On va se concentrer sur les deux premières propositions, puisque la troisième (augmenter la taille de la clé Kc) est correcte pour assurer un bon niveau de sécurité.

4.2.3.1 Première vulnérabilité et son remède

Le protocole EEMSUCU utilise la même clé Kc pour chiffrer l'IMSI et pour assurer l'intégrité du message de contrôle contenant le couple {$f10_{kc}$(IMSI), Rand}. Hors, selon la recommandation de NIST [Barker *et al.*, 2009] il est fortement conseillé de ne pas utiliser la même clé pour le chiffrement et l'authentification du message. Ceci peut aider les attaquants à détruire le château

de la sécurité. C'est pour cette raison, et comme nous avons vu, que la sécurité des réseaux UMTS et EPS utilisent toujours deux clés différentes pour la protection de l'intégrité et pour le chiffrement des communications (comme CK/IK et K_{NASenc}/K_{NASint}).

4.2.3.1.1 Remède proposé

Nous proposons d'utiliser une clé Kca de 256 bits, à la place de la clé Kc de 128 bits, afin de la diviser en deux parties égales : la première partie constituée par les 128 bits les plus signifiants forme la clé de chiffrement Kc destinée à chiffrer l'IMSI ; et la deuxième partie de la clé Kca constituée par les 128 bits restants, forme la clé d'authentification Ka destinée pour assurer l'intégrité du message {$f10_{kc}$(IMSI), Rand} en générant le code MAC en utilisant la fonction f12. Ce remède nécessite l'utilisation d'une nouvelle fonction f14 capable de générer 256 bits à la sortie, ou bien on peut utiliser la fonction f11 si elle est capable de générer les 256 bits nécessaires. Il se déroule comme suivant :

- Rand=f0() ;
- Kca =f14(K, Rand) ;
- Kc= (128 bits de gauche de Kca), et Ka=(128 bits de droite de Kca) ;
- Chiffrement (IMSI)= $f10_{kc}$(IMSI) ;
- MAC=f12(Ka, message).

Ou bien nous pouvons imaginer comme solution, de générer une nouvelle clé Ka, et exactement comme Kc, à partir de K pour l'authentification du message {$f10_{kc}$(IMSI), Rand}. Ceci peut être assuré en utilisant une nouvelle fonction f14 qui prend comme paramètres d'entrée Rand et K, et qui donne une clé Ka de 128 bits à la sortie :

- Rand=f0() ;
- Kc=f11(K, Rand) ;
- Ka=f14(K, Rand) ;
- Chiffrement (IMSI)= $f10_{kc}$(IMSI) ;
- MAC=f12(Ka, message).

4.2.3.2 Deuxième vulnérabilité et ses remèdes

Dans l'algorithme EEMSUCU, l'identification de l'abonné ne considère que les mobiles qui sont éteints et viennent de se mettre en marche en transmettant leur IMSI pour faire la demande d'attachement (*Attach request*) lors de la première connexion RRC. Or et comme nous avons vu l'abonné peut être en veille et le réseau de service, pour un certain problème où il ne peut pas récupérer son IMSI (à partir de son TMSI/GUTI), lui demande de transmettre son identité permanente.

Dans ce cas, si on veut assurer la sécurité de l'IMSI et le chiffrer en appliquant l'algorithme EEMSUCU, le HLR/AuC doit chercher non seulement dans les mobiles éteints (puisque le mobile n'était pas éteint) mais il faut essayer et tester toutes les paires (Ki et IMSIi) de tous les mobiles (éteints et allumés) qui sont stockés dans le HLR (UMTS) ou dans le HSS (4G) afin de trouver l'IMSIi (ou le MACi) correspondant et identifier le mobile. Autrement et si on ne fait pas ça, les abonnés légitimes dans ce cas seront expulsés du réseau. Cette procédure de recherche et d'identification est très lourde, elle introduit une charge sur les ressources de HLR/HSS et un

grand délai de traitement conduisant ainsi à une diminution de performance. Pour un HLR/HSS de $N=10^6$ abonnés par exemple, on a besoin de deux millions d'opérations (consommables avec f11 et f12) pour tester tous les utilisateurs. En moyenne 10^6 opérations sont nécessaires pour identifier un seul abonné dans le cas considéré. Ce nombre ne prend pas en compte la phase de comparaison entre les MAC.

4.2.3.2.1 Remèdes proposés

Pour éviter la recherche parmi tous les abonnés enregistrés dans un HLR/HSS et pour accélérer cette opération d'identification, nous proposons une solution qui permet de chercher seulement parmi les utilisateurs d'un VLR/MME donné. Ce qui réduit beaucoup la procédure de recherche et d'identification. Nous proposons un remède pour chacun des deux cas possibles où le réseau exige la transmission de l'IMSI d'un abonné. A rappeler, le premier cas se produit lorsqu'un abonné passe d'un réseau de service à un autre lors d'une mise à jour de localisation, et lorsqu'il y a un problème dans la connexion entre les deux réseaux de service. Le deuxième cas se produit lorsqu'un réseau de service tombe en panne et perd l'identité temporaire d'un mobile qui désire faire un appel ou effectuer une mise à jour de localisation à l'intérieur du même réseau en utilisant son identité temporaire.

Commençons par expliquer le remède proposé pour le premier cas et présenté dans la figure 4.5. Nous proposons qu'après la réception de l'IMSI chiffré de l'UE, le nouveau VLR/MME (VLRn/MMEn) envoie au HLR/HSS, à côté du message contenant l'IMSI chiffré, le numéro de l'ancien réseau de service VLR/MME (VLRa/MMEa) d'où vient l'utilisateur (voir figure 4.5). Le VLRn/MMEn connaît le numéro de VLRa/MMEa à partir du champ LAI/TAI (Tracking Area Identifier) existant dans le message de mise à jour de localisation envoyé par le mobile et incluant sa TMSI/GUTI. De cette manière, le HLR/HSS effectue sa recherche parmi les clés Ki des abonnés qui appartiennent seulement au VLRa/MMEa et non pas dans tout le HLR/HSS. Si le réseau téléphonique mobile dispose par exemple d'un million abonnés et de 5 réseaux de service, une recherche dans deux cent milles enregistrements appartenant à un VLR/MME donné doit être effectuée. Avec l'algorithme EEMSUCU on doit rechercher dans tout le million enregistrements que le HLR/HSS dispose. Le coût de recherche est donc réduit, dans cet exemple, par 5.

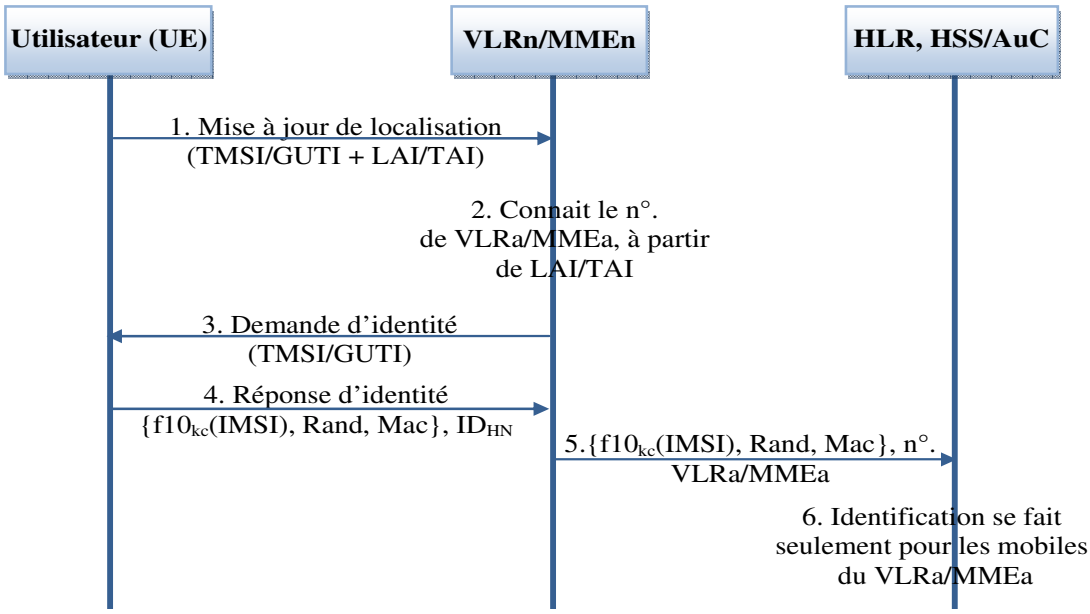

Figure 4.5. Recherche réduit après une mise à jour de localisation inter VLR/MME

Le même mécanisme s'applique dans le deuxième cas lorsque le VLR/MME perd l'identité temporaire TMSI/GUTI d'un abonné qui effectue un appel ou une mise à jour de localisation dans le même réseau de service.

Un abonné qui voudrait faire un appel, envoie normalement sa TMSI/GUTI au réseau, sans le numéro de zone où il se trouve. Pour l'algorithme EEMSUCU, si le réseau perd cette TMSI/GUTI et demande l'identité permanente, il provoque la recherche dans tout le HLR/HSS à la réception de l'IMSI chiffré. Pour ce cas, nous proposons que lorsque le VLR/MME demande de cet abonné d'envoyer son identité permanente, il lui demande également d'envoyer son LAI/TAI stocké dans son USIM. La réponse sera un message contenant à côté de l'IMSI chiffré, le LAI/TAI comme la figure 4.6 montre. Le VLR/MME à son tour, envoie lui aussi son propre numéro puisqu'il est responsable de la zone dont le code est le LAI/TAI reçu dans le message avant.

Presque la même démarche (de la figure 4.5) s'effectue pour le cas où l'abonné fait une mise à jour de sa localisation à l'intérieur du réseau de service où il se trouve. Dans ce cas, lorsque l'abonné envoie son TMSI/GUTI et le LAI/TAI au réseau de service courant, ce dernier sait bien que c'est lui-même qui gère cette zone. Ainsi ce VLR/MME envoie son numéro avec l'IMSI chiffré au HLR/HSS. C'est ainsi, en comparaison avec le cas précédent, le seul changement (voir figure 4.5) est qu'il n'y a plus un VLRn/MMEn et le seul réseau de service est le VLRa/MMEa courant (et pas l'ancien).

C'est seulement la demande d'attachement émise lorsque le mobile se met en marche qui déclenche dans notre proposition la recherche dans tous les mobiles éteints et le dernier message numéro 5 transmis dans ce cas ne contient aucune information supplémentaire sur le VLR/MME.

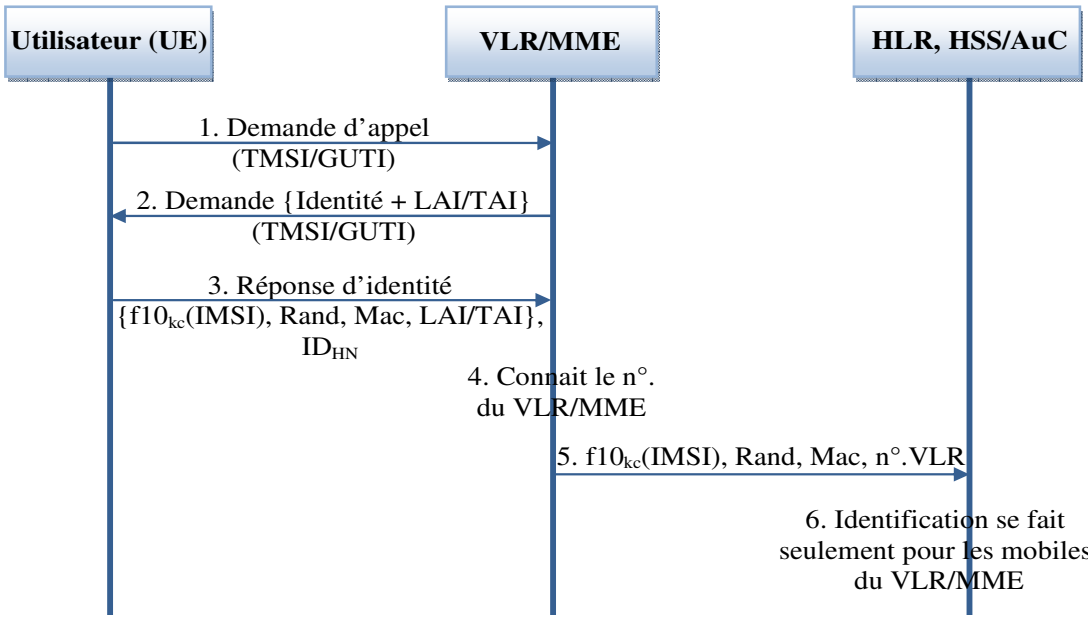

Figure 4.6. Recherche réduit après une demande d'appel

4.2.3.4 Attaques possibles et proposition d'amélioration

Après avoir présenté le protocole EEMSUCU, et pour mettre en lumière ses points faibles analysons les différentes attaques qui peuvent être effectuées sur le réseau :

a) Un attaquant qui écoute la liaison radio et qui arrive à capter la demande d'attachement contenant les paramètres {$f10_{kc}$(IMSI), Rand, Mac, ID_{HN}}, pourrait facilement envoyer ce message ultérieurement et plusieurs fois, et appliquer **une attaque par rejoue**. En recevant ce message, le VLR/MME ne vérifie et ne comprend que le champ ID_{HN} afin de savoir vers quel HLR/HSS il doit router la requête. Donc si le réseau de service reçoit plusieurs fois le même message, il ne s'en rend pas compte. Idem pour le réseau d'origine HLR/HSS. Ce dernier et à chaque fois qu'il reçoit le message, il fait la recherche de nouveau dans tous les mobiles afin de trouver le bon MACi et l'IMSIi correspondant. Donc une charge inutile des ressources peut s'effectuer dans le HLR/HSS, et cela peut le bloquer s'il dépasse sa capacité et ses limites.

b) Une modification dans les messages 'demande d'attachement' envoyés {$f10_{kc}$(IMSI), Rand, Mac, ID_{HN}} par un **attaquant au milieu**, et même avec l'existence du code MAC, va provoquer une longue recherche inutile et une occupation des ressources du HLR/HSS. Ce dernier n'a aucune idée de cette modification et ne découvre cela qu'après le calcul du code MACi qui exige la vérification de toutes les clés Ki des mobiles existants dans le réseau d'origine. Cette procédure provoque donc une recherche lourde et inutile qui peut causer un problème sérieux pour le HLR/HSS.

c) Un attaquant peut également créer des faux messages similaires de message d'attachement et les envoyer à un certain HLR/HSS qu'il vise à le faire tomber en panne (DoS attack). L'attaquant dans ce cas a besoin seulement de savoir l'ID_{HN} qui n'est pas difficile à connaitre. Cette identité constitue une partie de l'IMSI et plus précisément les digits les plus significatifs de l'IMSI comme nous allons expliquer après.

Toutes ces attaques ont un même objectif d'occuper les ressources du HLR/HSS dans le but de réaliser **une attaque de type déni de service**. Cela montre que le protocole EEMSUCU n'est pas assez robuste et n'assure pas un bon niveau de sécurité.

Par exemple avec 10^6 abonnés dans le HLR/HSS et 2 opérations (2 fonctions de hachage f11 et f12) pour la vérification d'une fausse demande reçue, le HLR doit effectuer $2*10^6$ opérations pour une simple requête. Supposons que f11 et f12 sont des fonctions de hachage SHA-256. Ceci veut dire que chaque fonction a besoin, d'après [Benchmarks, 2013], d'exécuter 15.8 instructions/octet. Pour la génération de 16 octets pour Kc par f11 ou les 16 autres octets pour le MAC par f12, ceci nécessite l'exécution de 252.8 instructions. Donc pour une requête d'IMSI, le HLR/HSS doit exécuter $505*10^6$ instructions/requête. Un attaquant (ou plusieurs attaquants) qui transmet plusieurs requêtes en même temps avec un taux de 120 requêtes/s par exemple, va forcer le HLR/HSS à exécuter $60600*10^6$ instructions/s. Si on choisit le serveur IBM le plus sophistiqué z196 pour fonctionner dans le HLR/HSS, ceci veut dire qu'il peut atteindre 52000 MIPS (Million Instructions Per Second) [IBM, 2013] dans une empreinte unique en offrant un total de 96 cores fonctionnant à 5.2 GHZ. L'attaque mentionnée force le HLR/HSS à exécuter plus de 60000 MIPS, et comme ceci dépasse sa capacité, le HLR/HSS se bloque (voire tombe en panne) et ne peut plus fonctionner. Cette vulnérabilité peut donc aboutir à une attaque de déni de service DoS.

Proposons maintenant une solution et une amélioration qui permet de résister contre ces types d'attaques.

4.2.3.4.1 Amélioration proposée

Pour éviter ces attaques, nous proposons qu'à la première réception de la demande d'attachement {f10kc (IMSI), Rand, MAC} et après la recherche dans les mobiles éteints et l'identification de l'abonné, le HLR/HSS stocke le Rand et le MAC correspondant à ce message dans le fichier de l'abonné qui contient son IMSI et tous les détails de son profil. Ensuite le HLR/HSS génère un numéro de séquence que nous appelons SQN_{RE} et l'envoie chiffré à l'utilisateur après la procédure AKA et après l'activation du chiffrement comme la figure 4.7 montre. Le rôle de SQN_{RE} est similaire au fonctionnement de TMSI/GUTI et sert à identifier l'abonné. Ce SQN_{RE} doit être stocké dans l'USIM de l'abonné afin de l'utiliser dans la transmission suivante de l'IMSI chiffré.

Durant la phase d'initialisation, si un attaquant transmet le même message envoyé avant (attaque par rejoue), c.à.d. avec le même Rand et le même MAC, le HLR/HSS va découvrir directement que c'est une attaque par rejoue en testant le Rand du message reçu (et pas en effectuant $2*10^6$ opérations) et ensuite le MAC (puisqu'on peut avoir quelques utilisateurs ayant le même Rand). Si le HLR/HSS trouve le même Rand accompagné du même MAC dans sa base de donnés, alors il rejette directement ce message en détectant que c'est une attaque par rejoue. Sinon le HLR/HSS va chercher seulement tous les mobiles éteints.

La procédure d'enregistrement du Rand/MAC et la génération du SQN_{RE} doit s'effectuer pour chaque abonné après sa première identification par le réseau. Donc on va arriver à un moment où tous les mobiles qui ont déjà fait la première *demande d'attachement,* possèdent un numéro SQN_{RE} stocké dans leur USIM et dans le réseau d'origine. Ainsi pour une nouvelle transmission d'IMSI chiffré, le mobile envoie son numéro SQN_{RE}, qui lui a été affecté comme suivant :

{$f10_{kc}$(IMSI), Rand, SQN_{RE}, MAC, ID_{HN}}. Le HLR/HSS teste seulement ce nouveau champ qui est capable d'identifier directement et facilement un mobile sans effectuer les deux opérations f11 et f12 sur tous les abonnés (les 10^6 enregistrements). Le champ SQN_{RE} permet d'effectuer une recherche très rapide dans la liste réduite des mobiles du VLR/MME ou éteints avec les deux opérations nécessaires pour identifier un abonné.

Un nouveau SQN_{RE} sera affecté au mobile après chaque émission d'IMSI chiffré. Pour ne pas garder la valeur du SQN_{RE} pour une longue durée, dans l'USIM et dans le HLR/HSS, et ouvrir la voie aux attaques possibles, nous proposons de changer régulièrement sa valeur affectée. A chaque allocation du TMSI/GUTI par le VLR/MME (mise à jour de localisation, etc.) ou à chaque procédure AKA, le VLR/MME contacte le HLR/HSS pour lui demander l'allocation d'un nouveau SQN_{RE} au mobile concerné afin de l'envoyer avec le TMSI chiffrés tous les deux.

Après l'implémentation et le fonctionnement de notre mécanisme proposé, un abonné qui vient de mettre son mobile sous tension, lors de la première connexion RRC, envoie le message {$f10_{kc}$(IMSI), Rand, Mac} pour s'attacher au réseau et se faire enregistré. Dans ce cas le HLR/HSS regarde le message, s'il ne contient pas un SQN_{RE}, il connaît que c'est un mobile qui établit une première connexion (ou c'est un message rejoué). Puis il effectue la recherche seulement sur les mobiles éteints et qui n'ont pas de SQN_{RE} afin de pouvoir identifier l'utilisateur. Par suite la recherche effectuée est réduite énormément.

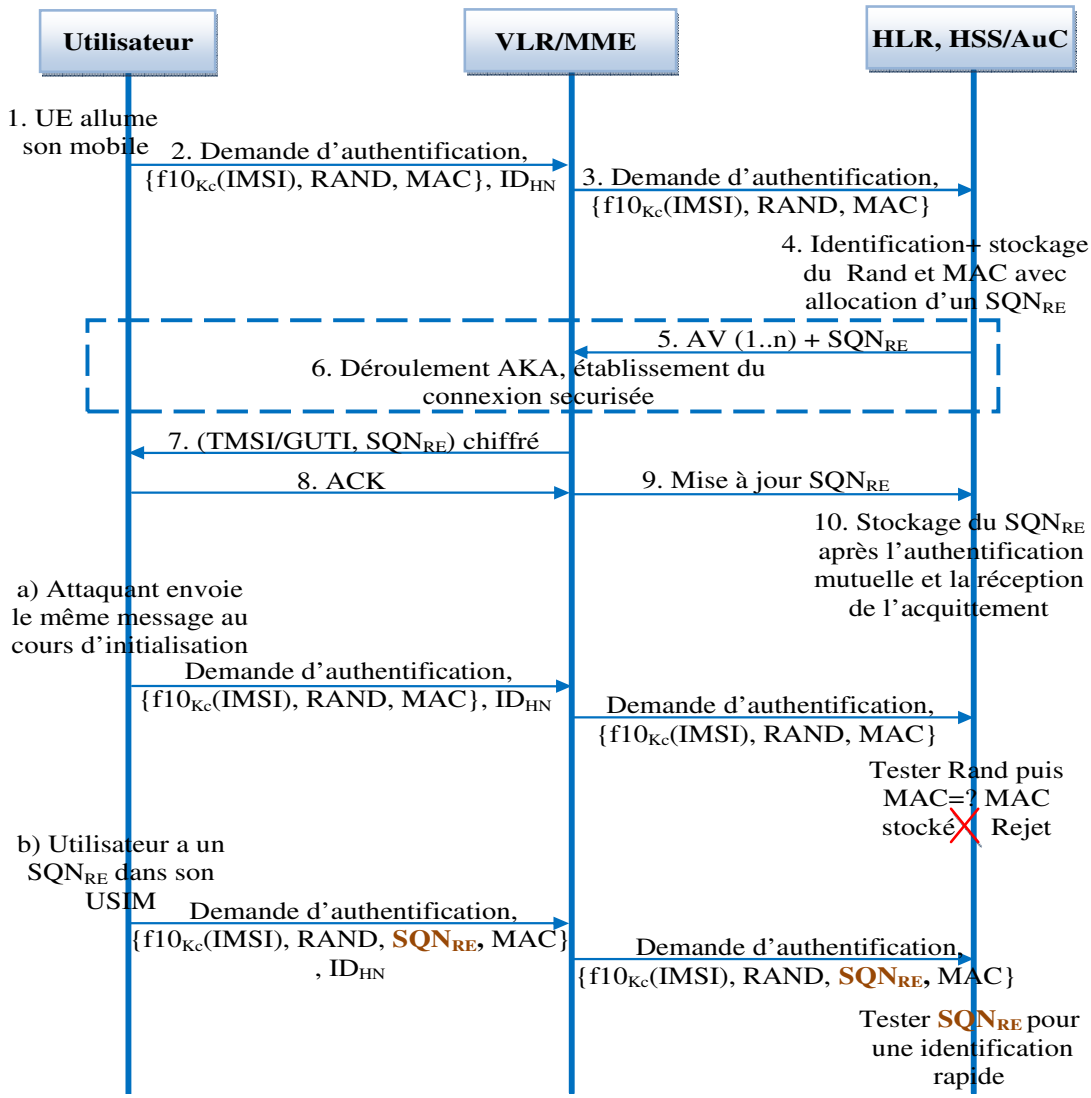

Figure 4.7. Détection rapide des attaques par le SQN_{RE}

Avec cette proposition nous nous bien défendons contre les trois types d'attaques qui provoquent l'attaque de déni de service du HLR/HSS puisqu'un message avec un ancien SQN_{RE} ou SQN_{RE} modifié se rejette directement après un simple test et une recherche rapide dans une liste réduite. Le rejet du demande se fait sans aucune opération (lorsque le SQN_{RE} n'existe pas dans la liste des enregistrements) ou maximum avec 2 opérations (lorsque le SQN_{RE} est identique par hasard à un autre déjà existant). Un message sans SQN_{RE} provoque la recherche dans une très petite liste des mobiles qui sont encore éteints et la charge n'est pas donc signifiante.

4.2.4 Comparaison des coûts d'identification d'un abonné

Pour résumer, nous présentons dans le tableau 4.1 une comparaison entre le protocole EEMSUCU et notre protocole proposé en termes de recherche effectuée et de nombre d'opérations n_{op} effectuées par le HLR/HSS afin d'identifier un abonné ou détecter une attaque.

Pour cela, nous considérons un réseau d'un million abonnés et cinq réseaux de service n_{VM} =5 (nombre de VLR/MME).

Nous avons vu dans notre protocole (avec nos remèdes et nos améliorations), qu'un abonné doit s'identifier toujours avec un SQN_{RE} qui lui a été attribué, et doit intégrer dans sa demande d'identification (dans certains cas) les champs nécessaires (comme le LAI/TAI) afin de réduire la recherche faite par le HLR/HSS. Par exemple, lorsque le réseau de service, et suite à un problème, demande l'identité IMSI de l'abonné, ou lorsque l'utilisateur met en marche son mobile. Dans ce cas et avec notre protocole, on effectue la recherche selon le SQN_{RE} de l'utilisateur soit dans les 200000 abonnés inscrits dans le réseau de service concerné soit dans la liste des mobiles éteints (par exemple 2500 dans le deuxième cas où l'utilisateur allume son mobile). C'est ainsi que le coût de notre protocole dans ces deux cas est de 3 opérations (en utilisant les fonctions f14, f12 et f10) afin d'identifier l'abonné au lieu d'une recherche dans 10^6 enregistrements avec 10^6 opérations pour l'EEMSUCU.

Coût d'identification d'un abonné	**HLR/HSS de N=10^6 abonnés avec n_{VM}= 5**		
	Scénario	**Enhanced EMSUCU (EEMSUCU)**	**Notre proposition (Recherche réduit, n_{op} négligeable et sans attaques)**
Envoie sécurisée de l'IMSI	Demande d'identité par le réseau de service	Recherche dans 10^6 enregistrements; n_{op} en moyenne=10^6	Recherche=2*10^5 enregistrements ; n_{op} = 3
	Utilisateur allume son mobile pour la première fois	Recherche=10^6 enregistrements ; n_{op} en moyenne=10^6	Recherche=2500 enregistrements ; n_{op} = 3
Attaques possibles	Attaque par rejoue	Recherche inutile=10^6 enregistrements ; n_{op} en moyenne=10^6	Recherche sur le SQN_{RE} n_{op} =0
	Attaque d'homme au milieu	Recherche inutile=10^6 enregistrements ; n_{op} =2*10^6	Vérification du SQN_{RE} n_{op} =0 ou 2 (coût d'EEMSUCU/10^6)
	Attaque de déni de service distribué sur le HLR/HSS	Avec 10 attaquants et 12 requêtes/s n_{op} =2.4*10^8	Avec 120 requêtes/s n_{op} ≤240 (coût d'EEMSUCU/10^6)

Tableau 4.1. Avantages de notre proposition en termes de coût de recherche et de nombre d'opérations n_{op} effectués par le HLR/HSS

Pour détecter une attaque, comme l'attaque par rejoue par exemple, notre protocole teste directement le SQN_{RE}. S'il n'est pas correct, il rejette directement la demande sans aucune opération. Pour l'attaque d'homme au milieu, si le SQN_{RE} modifié existe par hasard dans le HSS, ceci coûte deux opérations seulement pour découvrir que le code MACi est incorrect. Si la valeur de SQN_{RE} n'existe pas, ceci ne coûte aucune opération contre 2*10^6 opérations pour l'algorithme EEMSUCU qui teste tous ses abonnées un par un.

Prenons l'exemple d'une attaque de déni de service distribué avec plusieurs attaquants et 120 requêtes/seconde en total (10 attaquants et 12 requêtes/seconde envoyées par chacun). Comme nous avons vu, chaque requête force le HLR/HSS d'effectuer $2*10^6$ opérations. Alors ceci coûte $2.4*10^8$ opérations qui peuvent bloquer le HLR/HSS. Par contre pour notre protocole et avec 120 requêtes/s incluant des SQN_{RE}, où chaque requête nécessite au maximum 2 opérations, nous aurons au maximum 240 opérations ($n_{op} \leq 240$).

Après cette comparaison, il est clair que notre protocole proposé a un coût négligeable par rapport à l'EEMSUCU.

4.3 Analyse des vulnérabilités du protocole EPS-AKA

La procédure de l'EPS-AKA, et comme nous avons vu dans le chapitre précédent, se déroule avant l'établissement de la sécurité. Durant cette procédure, l'utilisateur et le réseau [TS 33.401, 2012] s'échangent des messages sans aucune protection comme les messages suivants : la demande d'authentification *Attach Request*, les messages AKA *User Authentication Request*, *User Authentication Response*, et les données d'authentification. Ceci ouvre la voie aux différents types d'attaques. Dans ce paragraphe nous allons effectuer une analyse et une synthèse des points faibles du protocole EPS-AKA et expliquer quels types d'attaques peuvent s'effectuer et quels dégâts cela peut causer. Nous pensons que 4 grands types d'attaques peuvent se faire contre des messages échangés ou des fonctions utilisées durant l'EPS-AKA. Expliquons-les et citons en deux mots le remède que nous proposons avant de présenter dans les deux paragraphes après, les solutions existantes et notre propre protocole proposé qui assure la protection contre tous ces types d'attaques.

4.3.1 Attaque de déni de service contre l'UE

Durant la procédure EPS-AKA, et comme l'UE et le MME ne peuvent pas authentifier les messages qu'ils échangent (ils ne possèdent pas encore des clés de sécurité communes), un attaquant peut modifier un message transmis ou renvoie un message déjà transmis. Ceci peut dans certains cas couper la connexion entre le mobile et le réseau et peut causer un déni de service. Une modification sur les capacités de sécurité de l'UE ou des messages d'authentification AKA [Khan *et al.*, 2008], ou la retransmission par un attaquant du message du rejet du mode de sécurité NAS (message de déconnexion), peuvent conduire à une attaque de type déni de service DoS sur l'UE. Expliquons comment ces attaques peuvent être effectuées.

4.3.1.1 Modification des capacités de sécurité d'UE

Les capacités de sécurité transmises par l'UE (UE Security Capability) au MME dans le message d'attachement, sont renvoyées à l'UE par le MME dans un message de réponse NAS (CMS NAS) protégé en intégrité [TS 33.401, 2012].

On appelle *bidding down attack* [Forsberg *et al.*, 2010] l'attaque qui modifie les capacités de sécurité de l'UE envoyé au MME. On ne peut pas détecter cette attaque qu'après la réception du CMS NAS par l'UE. Donc c'est après l'échange de cinq messages de signalisation qui provoquent une consommation de la bande passante et des ressources de calcul (pour la génération des AV(i), pour l'exécution de la procédure AKA, et la génération des clés K_{NASenc} et K_{NASint}), que l'UE reçoit le message CMS NAS contenant ses capacités modifiées. L'UE va ainsi

détecter cette différence entre ses capacités envoyées et celles reçues du MME. Il annule dans ce cas la connexion ou seulement la procédure d'attachement, et il envoie un message de rejet au MME. Si un attaquant effectue plusieurs fois l'attaque *bidding down attack* sur un certain utilisateur, ceci peut empêcher ce dernier de se connecter au réseau. Ce qui conduit à un déni de service pour cet abonné.

Le 3GPP recommande la protection contre l'attaque *bidding down attack* dans les deux sens. Pour cela nous pensons que la protection d'intégrité du message *Attach Request* (qui contient les capacités de sécurité) peut assurer la sécurité souhaitée puisque chaque entité réceptrice, MME ou UE, arrive dans ce cas à détecter directement cette attaque avant les échanges multiples des messages de signalisation.

L'UE annule la connexion et transmet un message du rejet lors de sa découverte d'une modification de n'importe quel paramètre du message CMS NAS. La solution consiste surement à chiffrer le message CMS NAS pour le bien protégé.

4.3.1.2 Attaque sur le message de rejet du mode de sécurité

Nous avons vu dans le chapitre précédent (paragraphe 3.5.4.4.2.1) que lorsque le MME envoie un CMS NAS à l'UE, ce dernier vérifie les capacités de sécurité reçus ainsi que l'intégrité du message en testant le code NAS-MAC. S'il y a un problème, l'UE envoie un message de rejet (*NAS Security Mode Reject*) afin d'annuler la connexion. Normalement, ce message de rejet est protégé avec les clés NAS établies dans une connexion précédente [TS 33.401, 2012]. Mais lors du premier attachement, ce message est envoyé sans aucune protection puisqu'il n'y a pas encore de contexte de sécurité NAS déjà établie. Ceci peut être un point faible et peut causer des problèmes pour les utilisateurs. En effet, un attaquant peut sauvegarder ce message de rejet et l'utiliser une autre fois après, pour empêcher un autre UE légitime d'accéder au réseau en lui coupant sa connexion. Ceci peut se faire plusieurs fois afin de provoquer un déni de service contre tous les utilisateurs qui viennent de mettre en marche leurs téléphones mobiles et qui désirent accéder au système EPS.

Les conséquences de cette attaque est la transmission de six messages de signalisation provoquant : la génération des AV(i), la procédure d'authentification, la génération des clés K_{NASint} et K_{NASenc} chez le MME et l'UE. Ceci peut aboutir à la fin à une coupure de la connexion. Pour cela nous pensons que la protection en intégrité et en confidentialité du message *NAS Security Mode Reject* peut être la solution à ce type d'attaque.

4.3.1.3 Modification des messages AKA (RAND, AUTN et RES)

Si l'UE reçoit une demande d'authentification (*Authentication Request*) contenant un RAND ou un AUTN modifié par un attaquant, il ne découvre pas cette modification qu'après les 4 étapes suivantes : la génération de la clé AK, le déchiffrement du SQN reçu par AK, la génération de XMAC, et la comparaison de XMAC avec le MAC reçu. Ces deux dernières valeurs ne sont pas égales bien sûr dans notre cas considéré où une des deux valeurs, RAND ou AUTN, est modifiée. En détectant la présence d'un problème, l'UE transmet un message d'échec d'authentification (*Authentication Failure*).

De même, si l'UE envoie une réponse d'authentification *(Authentication Response)* contenant un RES correcte, et cette réponse a été arrêtée et modifiée par un attaquant, le MME ne découvre pas la présence d'une attaque qu'après la comparaison de RES avec XRES, et après plusieurs opérations effectuées. Dans ce cas, le MME rejette la demande d'attachement et envoie à l'utilisateur un message de rejet d'authentification. En recevant ce rejet [Bouguen *et al.*, 2012], l'UE considère que, ou bien sa carte UICC est invalide ou bien que son terminal fonctionne mal et a besoin de se faire redémarrer (il doit l'éteindre et le remet en marche).

Si l'UE tente de s'attacher de nouveau et un attaquant modifie un de ses messages AKA, ceci provoquera un déni de service contre cet UE en l'empêchant d'accéder au réseau (à cause de l'échec de l'authentification du réseau/UE). Nous croyons que la solution de ce type d'attaque consiste à assurer la protection de l'intégrité des messages AKA.

Un attaquant qui sauvegarde une demande d'authentification valide envoyée par le réseau pour un certain utilisateur, peut utiliser cette demande pour provoquer un déni de service sur cet utilisateur lorsque ce dernier cherche à s'identifier de nouveau par son identité permanente IMSI. Lorsque l'attaquant envoie cette demande à l'utilisateur, il l'oblige à effectuer les 4 étapes que nous venons de citer, et il l'oblige également à générer le paramètre AUTS = $(SQN_{MS} \oplus AK) \parallel$ MAC-S. Ce dernier qui n'est que la concaténation des deux paramètres générés (par les fonctions f1* et f5*, voir annexe B1), est envoyé au MME dans un message d'échec de synchronisation. Nous voyons que la solution consiste à introduire des numéros de séquence pour éviter l'acceptation d'un ancien message du demande d'authentification (attaque par rejoue).

4.3.2 Attaques contre la clé secrète permanente K

Vue l'importance de la clé secrète K, si un attaquant arrive à avoir cette clé pour un certain abonné alors tout le château de sécurité de ce dernier s'écroule. L'attaquant est prêt à utiliser tous les moyens pour essayer de casser cette clé. Ceci a pour conséquences de compromettre toutes les communications suivantes et précédentes (si le trafic a été enregistré). Le seul moment, où la clé K est utilisée pour protéger les données qui peuvent être captées par un attaquant, est durant la procédure AKA.

La meilleure attaque contre la clé K consiste dans la cryptanalyse des fonctions de sécurité f1 à f5, utilisées durant la procédure AKA. Les messages qui sont transmis au cours de cette procédure peuvent être interceptés par un attaquant qui peut en profiter pour mener différents types d'attaques. Ces attaques de cryptanalyse sont classées en fonction du type du texte qui se trouve à sa disposition (voir paragraphe 1.2.6). Nous allons montrer les types d'attaques qui peuvent être montées contre chaque fonction de sécurité. Ceci permettra de mettre en évidence l'une des failles de la sécurité EPS, liées au fait que les fonctions de sécurité sont exposées à des attaques cryptographiques.

4.3.2.1 Attaque sur la voie radio

Lors du déroulement de la procédure AKA, les messages sont transmis en clair sans aucune protection. Dans ce cas, un attaquant qui écoute la voie radio peut facilement intercepter les messages échangés entre l'UE et le réseau. Ces messages peuvent être utilisés pour monter les attaques de cryptanalyse contre plusieurs fonctions de sécurité, dans le but de dévoiler la clé K.

Cette attaque est présentée dans la figure 4.8, où l'attaquant dispose des messages AKA : RAND, AUTN= (SQN⊕AK ‖ AMF ‖ MAC), et RES.

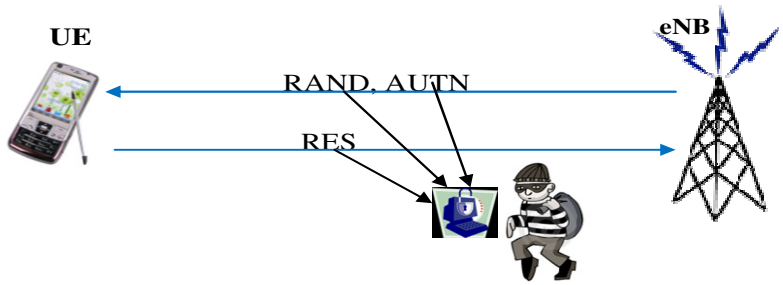

Figure 4.8. Attaque sur la voie radio

Ces messages (interceptés par l'attaquant) sont normalement utilisés dans les fonctions de sécurité f1 à f5 comme montre la figure 4.9.

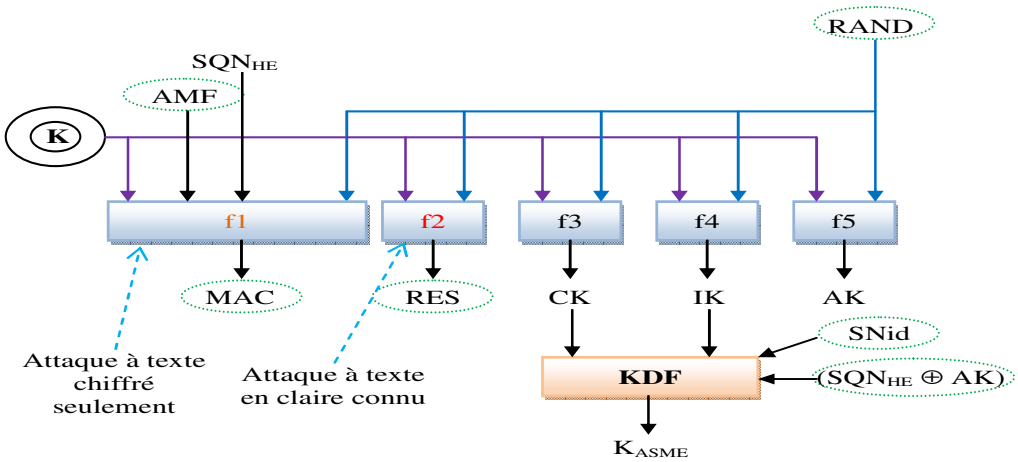

Figure 4.9. Exposition des fonctions de sécurité f1, f2 aux attaques cryptographiques

Les valeurs des paramètres interceptés RAND, AMF, MAC et RES sont encadrés par des cercles pointillés. Dans ce cas l'attaquant peut monter une attaque de type *à texte en clair connu* contre la fonction f2 [Caragata *et al*., 2011-b], puisqu'il possède son entrée RAND et sa sortie RES. Aussi il peut monter une attaque *à texte chiffré* contre la fonction f1, puisqu'il possède seulement sa sortie (en absence du SQN qui est masqué par AK). Donc la fonction 'f2' présente la faiblesse la plus grave parmi toutes les fonctions de sécurité utilisées. Pour bien protéger les fonctions f1 et f2 nous pensons que la solution consiste à chiffrer les messages AKA envoyés sur la voie radio et ne rien envoyer en clair. Dans ce cas l'attaquant doit commencer son attaque sur l'algorithme de chiffrement avec lequel RAND, AUTN et RES sont chiffrés.

4.3.2.2 Attaque contre la carte à puce UICC

Cette attaque contre la carte à puce est présentée dans la figure 4.10. Dans cette attaque [Caragata, 2011], le pirate utilise un ME modifié qui lui permet de voir les messages reçus et

envoyés par la carte UICC. L'attaquant utilise ces informations pour monter une attaque de cryptanalyse contre la clé K lors de la procédure AKA.

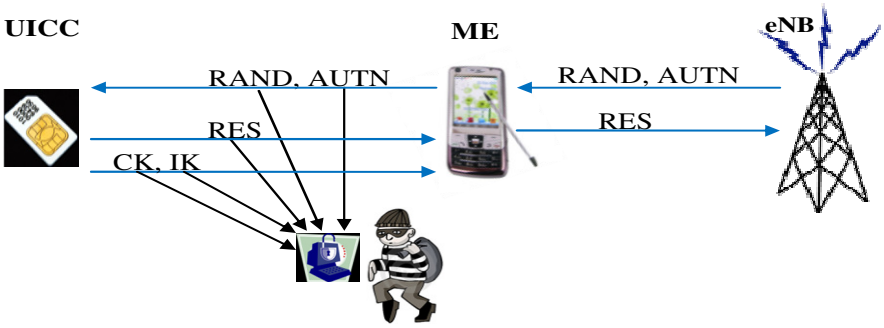

Figure 4.10. Attaque contre la carte à puce

La différence entre l'attaque sur la voie radio et cette attaque, vient du fait que la carte à puce transmet les clés CK et IK au ME. Alors, dans ce cas, l'attaquant peut monter des attaques *à texte en clair connu* contre les fonctions de sécurité 'f3' et 'f4' (avec l'attaque contre f2) comme montre la figure 4.11. Les données qui sont à sa portée sont encadrées par des cercles en pointillés. La clé K_{ASME} peut être également dévoilée si on connaît la fonction KDF utilisée.

La fonction f2 est donc la plus fragile et la plus exposée aux attaques. Les fonctions f1, f3, f4 sont aussi disposées à des attaques et c'est seulement la fonction 'f5', qui semble la plus protégée. Comme nous l'avons déjà mentionnés dans le chapitre 3 (paragraphe 3.3.2.2.5), il y a des méthodes de cryptanalyse qui permet de retrouver la clé secrète si on connaît la fonction utilisée comme la méthode qui a été développée par [Dunkelman *et al.*, 2010] pour retrouver la clé secrète de la fonction f8 cassée (KASUMI).

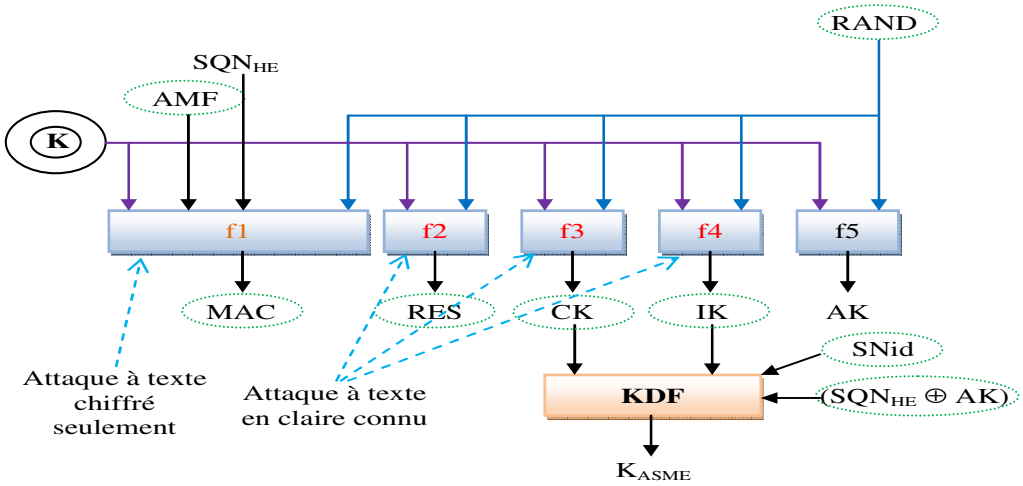

Figure 4.11. Attaques contre les fonctions de sécurité, via l'attaque contre la carte à puce

Notons que les attaques *à texte en clair choisi* ne sont pas possibles dans ce cas. En effet, un attaquant qui choisit et envoie les entrées RAND et AUTN, à la carte à puce, recevra forcement

un rejet par l'USIM après la vérification du code MAC. Par suite il ne réussira pas à découvrir aucun paramètre (RES, CK, IK) de la carte à puce.

Cette attaque (contre la carte à puce) est difficile à réaliser mais comme solution préventive, nous croyons que l'idéale est de compter sur la proposition [Caragata *et al.*, 2011-b] et qui consiste à augmenter la taille de la clé K à 256 bits et à utiliser une clé temporaire à la place de la clé K lors de la génération des AVs. Ceci limite l'exposition de la clé K puisque les données transmises sur la voie radio ne sont plus chiffrées par cette clé et le défi devient de casser la clé temporaire et non pas la clé permanente K.

4.3.3 Compromis des AV et blocage des services par un Attaque d'homme au milieu (MITM) entre MME et HSS

La sécurité de la liaison MME-HSS (complètement IP) n'est pas mentionnée directement dans les exigences de sécurité de la spécification [TS 33.401, 2012]. Cette dernière affirme seulement que cette liaison doit être protégée. Les spécifications techniques supplémentaires [TS 33.210, 2010] [TS 33.310, 2010] décrivent les différentes méthodes de protection des communications IP qui peuvent être utilisées entre les différents nœuds du réseau EPS. Elles conseillent d'utiliser l'IPSec ESP (voir paragraphe 3.5.3.4) pour les communications entre les nœuds, et elles laissent le choix libre à l'opérateur pour choisir la solution de sécurité convenable [Mjølsnes et Tsay, 2012]. Ces deux spécifications affirment que la protection de cette liaison est un problème interne pour chaque opérateur et que l'utilisation d'IPSec entre les entités est par conséquent optionnelle. Donc c'est l'opérateur qui décide le type de protection à utiliser entre MME et HSS.

En effet, une attaque de type MITM (Man in the Middle) sur la liaison entre le MME et le HSS n'est pas difficile à monter comme expliquée dans [Bou Abdo *et al*., 2013]. Cette vulnérabilité a été identifiée par [Bou Abdo *et al*., 2013] qui a compté sur le simulateur AVISPA (Automated Validation of Internet Security Protocols and Applications) [AVISPA Project, 2013] afin de montrer cette attaque. C'est pour cela, les échanges entre le HSS et le MME doivent obligatoirement être protégés, afin notamment d'empêcher la récupération des clés (la clé K_{ASME}) par un tiers malveillant. Pour cela nous croyons qu'il faut protéger les échanges entre les deux entités, MME et HSS, en confidentialité et en intégrité. Puisque l'attaque MITM peut provoquer :

a) le compromis des vecteurs d'authentification AV : Si les vecteurs d'authentification envoyés au MME sont dévoilés par l'attaquant (MITM), il pourrait se faire passer pour l'utilisateur et accéder au réseau compromis, ou bien il pourrait se faire passer pour n'importe quel réseau 4G et accéder aux informations de l'utilisateur en lui envoyant certaines requêtes et en utilisant RAND, AUTN et la clé K_{ASME}.

b) le blocage des services : Un attaquant qui modifie les messages *Authentication Data Request* ou *Authentication Data Response* entre MME et HSS, peut bloquer les services offerts par l'opérateur. Ceci peut être fait soit par le rejet de la demande par le HSS (si l'IMSI ou le SN id ont été modifiés), soit par la génération des AV incorrectes, soit par l'envoie de l'échec de l'authentification.

Pour cette raison et pour éviter ces problèmes nous croyons qu'il est fortement conseillé de protéger en confidentialité et en intégrité les demandes et les réponses des données d'authentification.

4.3.4 Attaques sur les réponses des données d'authentification (AVs)

Un autre point faible a été identifié dans le protocole EPS AKA par [Mjølsnes et Tsay, 2012]. Ceci a été faite après l'analyse de la sécurité du protocole par l'outil CryptoVerif [Blanchet, 2006]. Cette vulnérabilité peut être exploitée par un attaquant de l'extérieur ou de l'intérieur qui peut violer et permuter certains messages d'authentification échangés. Ces deux attaques, de l'extérieur ou de l'intérieur, s'appliquent même si les messages entre le MME et le HSS sont protégés en confidentialité et en intégrité. Ces attaques supposent que l'attaquant est en plein contrôle des messages envoyés entre l'UE et le MME et entre le MME et le HSS. Plus particulièrement, l'attaquant est capable de bien contrôler tous les messages entrants et sortants du MME.

On suppose aussi qu'il y a deux équipements utilisateurs, *U* et *U'*, qui lancent des sessions simultanées avec le même MME. Rappelons que lorsque MME envoie une demande des données d'authentification au HSS pour obtenir le vecteur d'authentification AV de l'UE (*U*), la réponse transmise par le HSS au MME est liée à l'UE (*U*). Ceci vient du fait que les paramètres qui forment AV sont générés à partir de la clé K de *U* (et partagée avec le HSS). Cependant, le MME ne peut pas vérifier pour quel utilisateur l'AV reçu a été généré, puisqu'il ne connaît rien sur la clé K partagée entre les utilisateurs et le HSS. Pour les deux attaques que nous allons expliquer l'attaquant profite bien de ce point là.

4.3.4.1 Attaque de l'extérieur

Le scénario et le flux des messages de cette attaque sont présentés dans la Figure 4.12. On suppose que les deux abonnés *U* et *U'* appartiennent au même HSS, et que tous les deux exécutent la procédure AKA avec le MME en même temps.

Au début, l'attaquant *A* laisse passer correctement les messages de signalisation contenant les identités (IMSI ou GUTI) des utilisateurs *U* et *U'* au MME et au HSS. Deux sessions AKA ont donc été lancées par le MME, une pour *U* et l'autre pour *U'*. Lorsque HSS envoie les réponses des données d'authentification AV de *U* et de *U'*, l'attaquant *A* permute (swap) ces messages de telle sorte que l'AV pour *U* (AV1) est circulé au MME comme étant la réponse par le HSS pour *U'*, et pareil pour l'autre réponse AV2 qui prend la place de AV1. Le MME ne se rend pas compte que les réponses ont été permutées l'une à la place de l'autre. Ensuite l'attaquant fais de nouveau une deuxième permutation entre les messages transmis de MME à *U* et à *U'*.

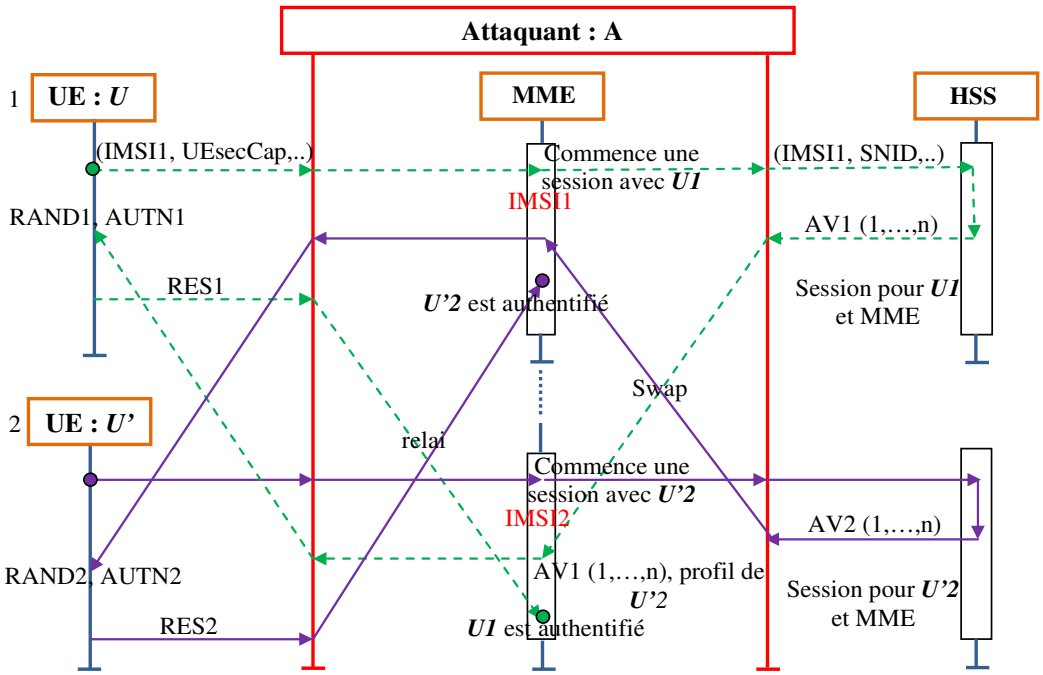

Figure 4.12. Attaque de l'extérieur sur EPS-AKA, où *U* est authentifié au MME en tant que *U'* et *U'* en tant que *U*

Donc *U* et *U'* reçoivent correctement les demandes d'authentification, contenant RAND1 et AUTN1 pour *U* et RAND2 et AUTN2 pour *U'*, qui leur ont été générées par HSS. Par conséquent, *U* et *U'* ne se rendent pas de l'attaque et envoient les bonnes réponses d'authentification (RES1 et RES2) au MME. Enfin, *A* permute encore une fois les réponses des deux utilisateurs *U* et *U'*. Il achemine la réponse de *U* au MME comme s'il vient de *U'*, et fait la même chose pour la réponse de *U'* (l'authentification de l'UE par le MME est clairement violée).

Dans cette attaque aucune entité, MME, *U* ou *U'* ne constate que *U* est authentifié auprès du MME en tant que *U'* et *U'* en tant que *U*. L'attaquant peut continuer la permutation (swap) des communications successives entre *U* et MME et entre *U'* et MME.

Comme conséquence, *U* utilise les services du réseau mobile au prix de *U'* et vice-versa. L'opérateur ne subit pas de perte financière immédiate. Mais si *U* et *U'* accumulent notablement des frais d'utilisation différentes, alors il y aura une perte financière pour l'un des utilisateurs, ce qui peut finalement coûter l'opérateur de HSS de perdre des clients, de l'argent et la confiance des abonnés.

4.3.4.2 Attaque de l'intérieur

Dans ce scénario, l'attaquant *A* est lui-même l'abonné *U* qui connaît l'IMSI d'un autre abonné honnête *U'* (en écoutant le réseau ou en utilisant l'IMSI catcher). Les deux appartiennent au même HSS, et l'attaquant *A* (ou *U*) effectue l'attaque présentée dans la figure 4.13 sans que *U'* soit présent.

L'attaquant lance simultanément deux procédures AKA, une de *U* et une autre de (l'IMSI de) U' en envoyant en même temps deux messages, un contenant sa propre identité IMSI1, et l'autre contenant l'IMSI2 de *U'*. Ensuite il effectue la même démarche et les mêmes permutations du scénario précédent. Le MME commence les deux sessions du protocole AKA, une pour *U* et une autre pour *U'*, mais il complète seulement la session de *U'* et l'autre session ne peut pas être complétée jusqu'à la fin puisque *A* ne connaît pas la clé K' de *U'*.

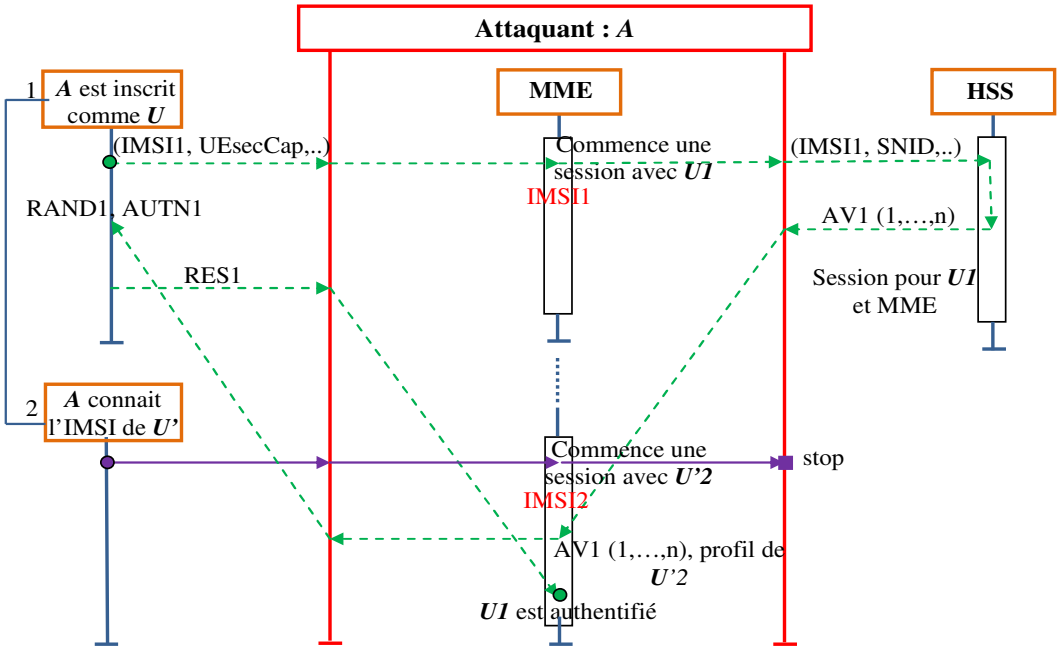

Figure 4.13. Attaque d'usurpation d'identité d'un abonné par un attaquant de l'intérieur

L'attaquant usurpe donc l'identité de *U'*, et il s'authentifie auprès de MME en tant que *U'*. Par ailleurs, l'attaquant connaît la clé K_{ASME} générée par le HSS pour *U*. Il est donc en mesure d'exécuter les étapes suivantes de communication, et d'utiliser les clés dérivées (NAS et AS) pour protéger les services mobiles fourni par le MME au nom (sur le compte) de *U'*.

Comme conséquence, *A* (*U*) utilise les services du réseau offerts pour l'utilisateur honnête *U'* et ce dernier sera facturé pour ces services. En effet, l'attaque ne produit aucun frais supplémentaire pour *A* mais ca peut lui causer des problèmes. Par exemple, si *A* commet des crimes à partir des endroits où *A* se connecte à chaque fois via un eNB en tant que *U'*, ce dernier peut devenir injustement un suspect quand les responsables utilisent les données de connectivité au réseau pour faire une enquête sur ces crimes.

4.3.4.3 Remède contre ces attaques

La vulnérabilité qui conduit à ces deux attaques vient de la permutation (swap) non détectée des réponses des données d'authentification. Cette vulnérabilité peut être contrecarrée si le MME peut déterminer pour quel IMSI la réponse du HSS est générée.

Comme solution nous pensons que le MME doit générer et utiliser un identificateur de la session, *Session-id* pour chaque demande des données d'authentification envoyée au HSS. Ce dernier doit

à son tour inclure cet identificateur de session ainsi que la valeur de l'IMSI de l'utilisateur reçu dans la réponse des données d'authentification. De cette manière, le MME possède tous les indications nécessaires pour vérifier avec certitude à quel UE (IMSI) l'AV (reçu de HSS) appartient. Il pourrait donc attribuer facilement chaque réponse à sa demande correspondant. Pour éviter les attaques actives (par exemple la modification de *Session-id* ou d'IMSI) sur les demandes et les réponses des données d'authentification, la protection en intégrité et en confidentialité de tous les messages échangés entre MME et HSS est obligatoire.

4.4 Protocoles existants et proposés pour remplacer l'EPS-AKA

Différentes variantes du protocole AKA ont été proposées afin de renforcer la robustesse du protocole EPS-AKA [Xiehua *et al.*, 2011] [He *et al.*, 2008] [Bou Abdo *et al.*, 2012-a] [Cho *et al.*, 2012] et afin de se protéger contre les multiples attaques possibles. Il y a des chercheurs qui ont proposé des améliorations sur le protocole existant [Xiehua *et al.*, 2011] [Bou Abdo *et al.*, 2012-a] [Cho *et al.*, 2012] et d'autres ont proposé de le changer carrément et d'utiliser un nouveau protocole d'authentification mutuelle et d'établissement des clés [He *et al.*, 2008].

Nous allons étudier les deux protocoles les plus importants SE-AKA (Security Enhanced-AKA), et EC-AKA (Ensured Confidentiality-AKA) qui ont été proposés récemment pour remplacer le protocole EPS-AKA. Nous allons analyser ces protocoles pour estimer leur robustesse et leur capacité à résister contre les attaques. Comme le 3GPP recommande [Forsberg *et al.*, 2010] d'utiliser la cryptographie à clé publique pour protéger l'IMSI, ces deux protocoles sont basés sur une infrastructure à clé publique afin de sécuriser les messages échangés.

4.4.2 Protocole SE-AKA

Le protocole SE-AKA [Xiehua *et al.*, 2011] a introduit des améliorations importantes sur l'EPS-AKA. Il a sécurisé la transmission entre les différents nœuds du réseau EPS par la protection, via le chiffrement à clés asymétriques, de presque tous les messages échangés entre les différentes entités du réseau. Les clés publiques utilisées doivent être implémentées par les nœuds sous forme de certificats électroniques avant le début de l'exécution du protocole proposé. Dans ce dernier, l'IMSI, les messages échangés entre MME-HSS et le message de la demande d'authentification, sont tous chiffrés par des clés asymétriques comme montre la figure 4.14. Décrivons maintenant le protocole et analysons-le.

L'UE commence l'exécution du protocole SE-AKA par le chiffrement de son IMSI à l'aide de la clé publique du HSS (PK_H) partagée par tous les abonnés du réseau d'origine et qui est sauvegardée dans la carte UICC de chaque utilisateur. L'UE envoie la demande d'accès contenant son IMSI chiffré, $A=\{IMSI\}_{PKH}$, et l'identité du HSS ID_{HSS} auquel l'utilisateur appartient et vers lequel le MME doit router la demande. À la réception de cette demande par le MME, ce dernier utilise également le PK_H pour chiffrer son identité du réseau SN id pour générer le paramètre $B=\{SN\ id\}_{PKH}$. Puis, la demande des données d'authentification formée par A et B, sera envoyée au HSS. À la réception de cette demande, le HSS déchiffre A et B par sa clé privée pour extraire l'IMSI de l'utilisateur et le SN id. Ensuite il envoie par un message C, les *n* vecteurs d'authentification AV et l'IMSI de l'abonné, chiffrés tous par la clé publique PKM du réseau de service MME. Ce dernier transmet la demande d'authentification à l'UE, comme un message D chiffré par la clé publique de l'abonné. Le message D contient les paramètres : RAND, SN id,

l'identité temporaire S-TMSI, et l'identité de clé K_{ASME} établie KSI_{ASME}. La suite du protocole se fait exactement comme dans l'EPS-AKA sauf le dernier message E transmis chiffré, de l'UE au MME, par la clé publique de ce dernier.

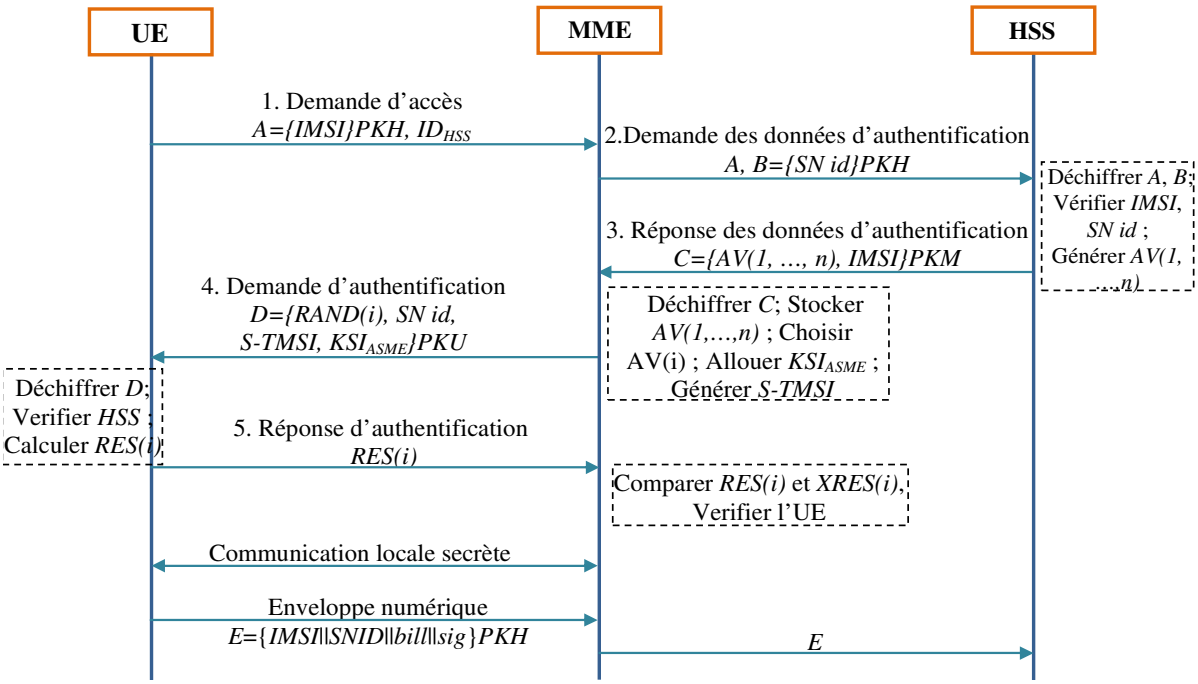

Figure 4.14. Messages de signalisation SE-AKA

4.4.2.1 Cryptanalyse du protocole SE-AKA

Comme l'identité IMSI d'un abonné est toujours fixe et ne change pas, par suite son chiffrement par la même clé publique (du HSS), et en utilisant la méthode RSA ou ECC comme proposé par [Xiehua *et al.*, 2011], donnera toujours la même valeur qui ne change jamais. Ainsi un utilisateur 'U' qui transmet son IMSI ou sa valeur chiffrée qui est aussi constante tout le temps, aura la même confidentialité et la même robustesse vis-à-vis les attaques contre l'IMSI et le chiffrement n'améliore pas le niveau de sécurité. Il suffit de dévoiler et suivre la trace de l'IMSI chiffré pour connaître et identifier un certain utilisateur. Un IMSI chiffré statique et employé tout le temps le même, sera aussi vulnérable que le IMSI lui-même. Le chiffrement de l'IMSI n'apporte donc rien au niveau de sécurité. Ceci peut aboutir à une attaque par dictionnaire. D'autres attaques peuvent aussi être montées sur SE-AKA. Nous avons analysé le protocole SE-AKA et nous avons trouvé qu'il est vulnérable à plusieurs attaques : attaque par dictionnaire, attaque par rejoue, attaque de déni de service sur le HSS/AuC et sur l'UE, et attaque MITM.

4.4.2.1.1 Attaque par dictionnaire sur l'IMSI chiffré dans SE-AKA

Avant d'expliquer comment l'attaque par dictionnaire sur l'IMSI chiffré peut être montée, rappelons tout d'abord la structure de l'identité IMSI. Cette dernière suit le plan d'identification E.212 de l'UIT, et elle est composée de trois champs qui ne dépassent pas les 15 digits et ils sont les suivants (voir figure 4.15) :

- MCC (Mobile Country Code) : indicatif du pays domicile de l'abonné mobile (par exemple 415 pour le Liban et 208 pour la France) ;
- MNC (Mobile Network Code) : indicatif du réseau mobile de l'abonné (au Liban 03 pour 'Touch', et en France 01 pour 'Orange') ;
- MSIN (Mobile Subsriber Identification Number) : numéro de l'abonné mobile à l'intérieur du réseau.

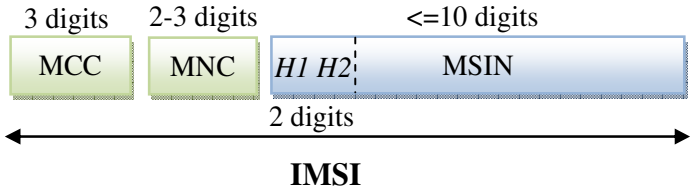

Figure 4.15. Composition de l'IMSI

Les 5 premiers digits sont connus au public et permettent de déterminer, de façon unique dans le monde, le réseau mobile de l'abonné. Les 2 premiers digits, appelés H1 H2, du champ MSIN donnent l'indicatif du HSS de l'abonné au sein de son réseau (en cas où il y a 2 HLR ou plus). Le MME est donc capable, à partir d'un IMSI quelconque de connaître le HSS convenable. Pour cela lorsque l'UE chiffre l'IMSI, le MME ne peut plus déduire quel est le bon HSS vers lequel il doit router les messages. Il est donc nécessaire d'informer le MME de l'ID du HSS pour qu'il puisse lui adresser les messages de signalisation.

Le tableau 4.2 montre la partie constante de l'IMSI pour les opérateurs mobiles du Liban. Le nombre possible de toutes les identités possibles IMSI pour un certain opérateur est donc 10^{10} ou bien 10^{8} en présence de l'indicatif du HSS. Pour le protocole SE-AKA, H1 et H2 peuvent être facilement découverts en interceptant l'ID_{HSS} envoyé en claire sur la voie radio dans le premier message de demande d'attachement. Même pour l'EPS-AKA, ils peuvent être connus également en achetant une ou plusieurs cartes UICC du marché d'un opérateur et en utilisant un lecteur IMSI (USB SIM Card Reader).

Tableau 4.2. Gamme des IMSI pour les opérateurs du Liban

MCC	MNC	Première partie de l'IMSI	Nom de marque	Opérateur	Etat
415	01	41501	Alfa	MIC1/Orsacom Telecom	Opérationnel
415	03	41503	Touch	MIC 2/ Zain	Opérationnel
415	05	41505	Ogero Mobile	Ogero Telecom	Planifié pour le futur

Par exemple, chaque IMSI de l'opérateur libanais 'Touch' a les 7 premiers digits égaux à 41503*H1H2* et l'abonné sera identifié par les 8 derniers digits. Le système de chiffrement à clé publique le plus important et le plus utilisé parmi les algorithmes asymétriques est le RSA. Pour effectuer une attaque sur l'IMSI transmis dans le SE-AKA, il faut chiffrer par le RSA toutes les IMSI possibles par la clé publique PKH connue, et mettre le résultat dans un tableau. En captant

la demande d'accès qui contient l'IMSI chiffré de l'UE, tout ce qu'on doit faire est de chercher dans le tableau et de trouver la même valeur afin de découvrir le vrai IMSI. Nous avons effectué la simulation de l'attaque par dictionnaire en calculant toutes les valeurs possibles de l'IMSI de 'Touch', et en utilisant une clé publique RSA de 1024 bits.

Dans cette simulation, nous avons calculé deux gammes des IMSI de 41503ab00000000 au 41503ab99999999 et de 41503cd00000000 au 41503cd99999999.

La taille de ce tableau contenant toutes les IMSI possibles de l'opérateur 'Touch' et ses IMSI chiffrés par une clé publique est de 27.1 GB. Le temps de calcul nécessaire pour remplir ce tableau est de 42 heures pour un PC qui a les caractéristiques suivants : Core 2 Duo CPU, P8600@ 2.4GHz, 4 GB du RAM, Windows 7, 1 HDD NTFS 5400 rps. Si on utilise les courbes elliptiques ECC à la place de RSA, la taille du tableau sera réduite par un terme de 3.5 (avec une clé ECC de 256 bits) mais le temps de calcul augmente.

Possédant ce tableau bien rempli, une fois un utilisateur envoie (une demande d'attachement) une valeur chiffrée de son IMSI, il suffit de trouver cette valeur dans le tableau pour dévoiler l'identité IMSI de l'abonné.

Pour considérer un protocole de sécurité comme sécurisée ou sûr, le temps de cryptanalyse doit être supérieur à 10^{17} ans en utilisant l'ordinateur le plus rapide au moment où le test commence. Le temps nécessaire pour remplir le tableau pour pouvoir découvrir l'IMSI de n'importe quel abonné est largement inférieur à 10^{17} années, donc SE-AKA est considéré comme non sécurisé.

De plus si on imagine que l'opérateur ne change pas souvent les clés publiques de ses nœuds, HSS et MME, donc on peut imaginer qu'une fois nous avons rempli le tableau, la découverte des IMSI après ne prend qu'un laps de temps. Et dans ce cas l'attaque sera appelée attaque par dictionnaire.

En pratique un opérateur n'utilise pas tous les 10^8 IMSI possibles, mais une partie de ces IMSI (de l'ordre de million). Les IMSI utilisés pour un opérateur sont généralement partagés dans des groupes. D'après [Bou Abdo *et al.*, 2012-a] on peut estimer ces groupes avec un haut niveau de confidence en recueillant un certain nombre des cartes SIM du marché et à partir desquelles nous extrairons leur IMSI. Ceci nous permet d'estimer 80% de ces IMSI actifs [Bou Abdo *et al.*, 2012-a] d'une façon très rapide et sans attendre des heures et des heures pour remplir tout le tableau de tous les 10^8 IMSI. ;

Pour éviter ce type d'attaque, nous pensons qu'il vaut mieux utiliser soit l'algorithme asymétrique ELGAMAL [El Gamal, 1985] soit l'algorithme à clé publique RSA-OAEP [Jonsson et Kalisiki, 2003]. Ces deux algorithmes ont la caractéristique de donner des sorties différentes même si le texte en clair à chiffrer et la clé publique restent les mêmes et ne changent pas. L'inconvénient de l'algorithme El Gamal est qu'il est moins performant que RSA, et la taille du texte chiffré à la sortie est le double de la longueur du texte en clair (et de RSA) et par suite il occupe plus la bande passante sur la voie radio et sur la liaison entre le MME-HSS. L'algorithme RSA-OAEP est très sûr. Il ajoute des données pseudo-aléatoire au texte en clair à chaque processus de chiffrement pour résoudre le problème d'éviter les sorties statiques pour une même entrée.

4.4.2.1.2 Différentes attaques possibles sur le SE-AKA

Dans le protocole SE-AKA, un attaquant peut intercepter la demande d'attachement, A={IMSI}PKH, ID_{HSS}, et le renvoyer après pour effectuer l'**attaque par rejoue.** Ceci provoque : le chiffrement asymétrique de SN id par le MME, le déchiffrement de A et B par la clé privée de HSS, le calcul de *n* vecteurs d'authentification AV (1..n) par le HSS/AuC (la phase la plus coûteuse avec la phase suivante), le chiffrement asymétrique des AV(1..n) et de l'IMSI au niveau du HSS (par la clé publique du MME, PKM), le déchiffrement de C, et la transmission de la demande d'authentification sur la voie radio par le MME (chiffré par la clé publique de l'UE, PKU). C'est une consommation inutile des ressources de calcul surtout par le HSS/AuC et une occupation inutile de la bande passante sur la voie radio et sur la liaison MME-HSS. Ceci peut conduire à une attaque de déni de service DoS contre le HSS/AuC. Si un (ou plusieurs) attaquant(s) envoie(nt) plusieurs demandes d'attachement déjà interceptées, au HSS/AuC (DDoS), ou s'il utilise le tableau qui contient les IMSI actifs (et chiffrés) d'un opérateur, et lance un logiciel qui les envoie automatiquement, il peut ainsi monter une attaque de **déni de service contre le HSS/AuC.**

Un blocage des services peut se faire aussi par **une attaque MITM** de la façon suivante. Un attaquant qui modifie le SN id peut provoquer le rejet de la demande par le HSS. Si l'attaquant remplace $B=\{SN\ id\}_{PKH}$ par un autre $B'=\{SN\ id'\}_{PKH}$ légal (autorisé par le HSS), il provoquera la génération des AV (1..n) incorrectes (pour le réseau qui a SN id'). Le message B peut être facilement distingué par l'attaquant puisque le SN id et le PKH sont des valeurs publiques. L'échec de l'authentification est encore possible si la modification atteint le message C. En outre, un MITM peut usurper l'identité de (se faire passer pour) MME ou HSS. Tous ces inconvénients proviennent de l'absence de l'authentification de la source et de l'intégrité de ces messages.

Enfin on a une attaque de **déni de service contre l'UE**, par la modification du message RES envoyé en clair.

4.4.3 Protocole EC-AKA

Ce protocole présenté [Bou Abdo *et al.*, 2012-a] dans la figure 4.16, apporte beaucoup des améliorations importantes sur l'EPS-AKA. Dans ce protocole, presque tous les messages sont protégés en intégrité et en confidentialité. Il utilise le chiffrement asymétrique pour chiffrer les messages *A*, *B*, et *C* en se basant sur les clés publiques de HSS (*PKH*), et de MME (*PKM*), et sur le chiffrement symétrique pour chiffrer les autres messages *D*, *E* et *F* en se basant sur la clé de chiffrement *EK* générée dans l'UE et dans le HSS, et envoyé par ce dernier au MME. L'algorithme de chiffrement/déchiffrement symétrique adopté (*chosen UESecCapability*) avec la clé *EK,* est choisi par le MME à partir des capacités de sécurité reçus (*UE Sec Cap*) dans le message *C*. Pour plus de sécurité, l'algorithme choisi est masqué par la valeur aléatoire (*RandomUEsecCapab1*) transmis par l'UE au début dans le message *A*.

En effet, l'UE n'envoie pas seulement son IMSI et les capacités de sécurité (*UESecCapabilities*) dans la demande d'attachement, mais il transmet aussi des valeurs aléatoires comme le *RandomEncKey*, et le *RandomIntKey* qui servent à garantir l'obtention d'un IMSI chiffré dynamique. Ce qui permet de protéger l'IMSI contre l'attaque par dictionnaire. Ces valeurs aléatoires seront utilisées aussi pour d'autres fins de chiffrement et d'intégrité.

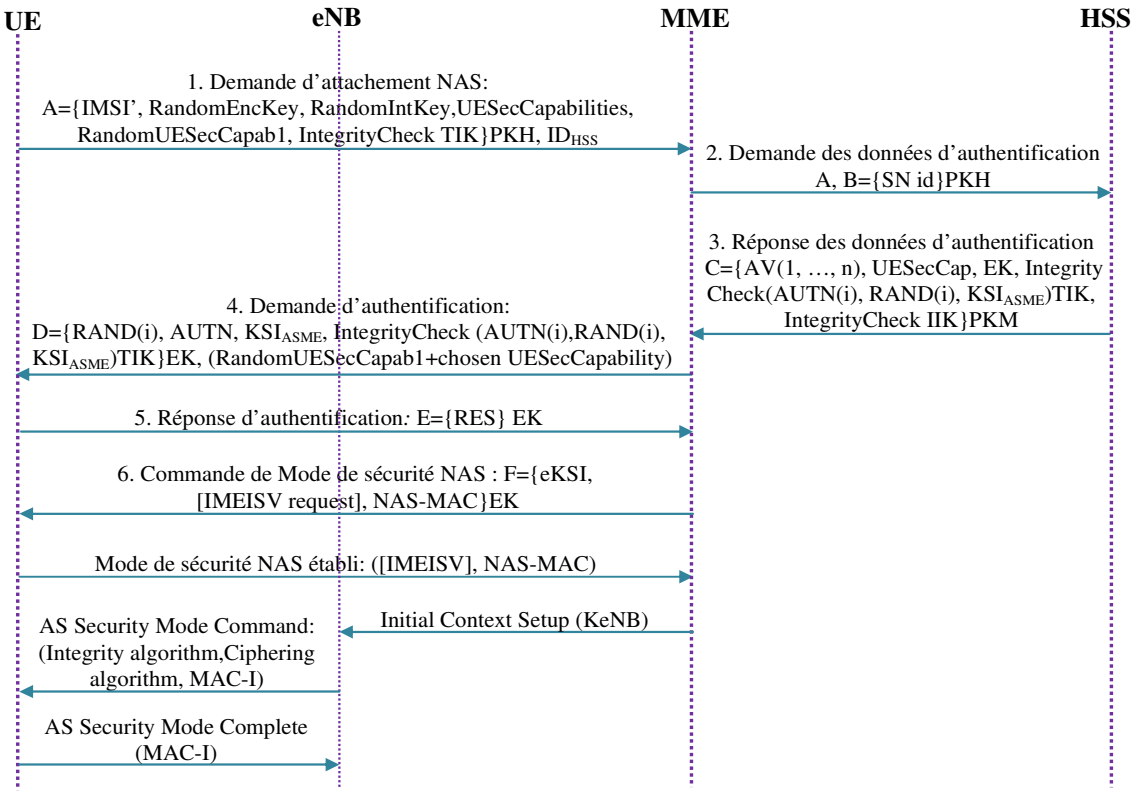

Figure 4.16. Messages de signalisation du protocole EC-AKA

La sécurité de ce protocole est basée sur deux clés pré-partagées entre l'UE et le HSS, appelées PIK (Permanent Integrity Key) et PEK (Permanent Encryption Key). Ces dernières sont des clés intermédiaires utilisées avec les deux valeurs aléatoires *RandomIntKey* et *RandomEncKey* afin de générer respectivement la clé d'intégrité *TIK*=PIK||*RandomIntkey*, et la clé de chiffrement *EK*=PEK⊕*RandomEncKey* au niveau de l'UE et du HSS. Ceci garantit d'avoir toujours des clés de sécurité dynamiques qui changent avec chaque attachement de l'abonné.

La clé *TIK* consiste à protéger : l'intégrité du message *A* avant le chiffrement, et l'intégrité des paramètres (AUTN(i), RAND(i), KSI_{ASME}) envoyé dans les messages *C* et *D*. Notons que les notations *IntegrityCheck (M)x* et *M, IntegrityCheck x* existant dans les messages 1,3 et 4 représentent le code d'intégrité MAC appliqué sur le message *M* en utilisant la clé d'intégrité *x*.

4.4.3.1 Cryptanalyse du protocole EC-AKA

Les améliorations de l'EC-AKA ont ajouté une sécurité importante au réseau, mais en analysant ce protocole nous avons trouvé qu'il est vulnérable aux trois attaques : attaque par rejoue, attaque de déni de service sur le HSS/AuC, et l'attaque sur les réponses des données d'authentification.

1) **Attaque par rejoue** : Un attaquant qui écoute le canal radio, et qui envoie le même message {A, ID_{HSS}} envoyé avant, va provoquer les conséquences suivantes:

a) Le HSS doit effectuer : trois opérations de déchiffrement asymétriques (deux par la clé privée *KRH* du *HSS*, et une par *PKM* pour la vérification de la signature), une génération des clés *TIK* et *EK*, une vérification de l'intégrité du message *A*, une génération de *n* vecteurs d'authentification,

une génération de l'*IntegrityCheck* (AUTN(i), RAND (i), KSI_{ASME})TIK avec la clé *TIK*, une génération de l'*IntegrityCheck* sur le message *C* par la clé IIK, et un chiffrement asymétrique de *C*.

b) Une allocation des ressources des nœuds du réseau cœur, ainsi que la bande passante entre HSS et MME.

c) Au niveau du MME, un déchiffrement asymétrique de *C* (plusieurs blocs RSA) doit être effectué, avec une vérification d'intégrité par la clé IIK. Le MME continue par l'émission sur la voie radio de la demande d'authentification. Puis, l'attaquant envoie le RES chiffré (stocké avant par l'écoute) pour faire appliquer encore un déchiffrement, au niveau du MME, du RES qui est incorrecte. Dans EC-AKA le HSS ne détecte pas cette attaque et n'a aucune mesure pour l'éviter.

2) **Déni de service sur le HSS/AuC** : Un grand nombre de ces demandes envoyées au HSS/AuC peut paralyser le HSS et l'empêcher à faire son travail.

3) À la réception du message *C*, le MME ne peut pas connaître à quel utilisateur les vecteurs d'authentification reçus appartiennent. Il n'y a aucune indication qui peut l'aider à connaître le destinataire. Comme nous venons de voir avec le protocole SE-AKA, **l'attaque sur les réponses des données d'authentification** peut se produire facilement dans ce cas.

4) Un attaquant peut **usurper l'identité de MME** en remplaçant B par un autre message B' statique contenant un SN id' autorisé, signé par *PKM* et chiffré par *PKH*. Dans ce cas il se passe pour un MME et causer des dégâts comme nous avons expliqué avant.

Dans les deux protocoles SE-AKA et EC-AKA, si quelqu'un arrive à avoir la clé privée du MME il peut facilement connaître tous les vecteurs d'authentification (ainsi que les identités IMSI, et les clés EK dans EC-AKA) envoyés, avant et dans le futur, par le HSS. Par suite il peut avoir les mêmes conséquences, déjà expliquées, de compromis des AV par un MITM.

4.5. Notre protocole proposé FP-AKA

Tous les protocoles d'authentification mutuelle et d'établissement des clés existants et déjà proposés dans la littérature, présentent certaines vulnérabilités contre plusieurs attaques. Nous venons d'analyser deux protocoles récents et performants et nous avons montré comment ils présentent des points faibles. Nous allons proposer dans ce paragraphe, notre propre protocole que nous appelons FP-AKA (Full Protection- Authentication and Key Agreement). Ce protocole est inspiré de l'EPS-AKA et des deux protocoles EC-AKA et SE-AKA. Il regroupe leurs avantages et il introduit des améliorations afin de résoudre toutes les faiblesses possibles. Il assure la protection contre toutes les attaques déjà étudiées et qui ont réussi à fragiliser les deux protocoles analysés EC-AKA et SE-AKA. Notre protocole a été vérifié par l'outil de vérification de la sécurité des protocoles, AVISPA, qui a indiqué FP-AKA comme un protocole très sûre. L'implémentation de FP-AKA sur l'AVISPA [AVISPA Project, 2013] a été faite en utilisant le langage HLPSL (High-Level Protocol Specification Language) [Glouche, 2008].

Avant de présenter le protocole proposé, définissons les paramètres essentiels qui nous seront utiles pour bien expliquer et bien détailler le FP-AKA.

4.5.1 Nomenclature

- **PSQN** (32 bits) : une valeur aléatoire de 32 bits spécifiée par l'UE lors de la première connexion. À partir de la deuxième connexion cette valeur sera égale au NSQN (New SQN) fourni par le HSS.

- **RandIK** (128 bits) : une valeur aléatoire (Random Integrity Key) générée par l'UE et utilisée pour dériver la clé d'intégrité TIK.

- **RandEK** (128 bits) : une valeur aléatoire (Random Encryption Key) générée par l'UE et utilisée pour dériver la clé de chiffrement EK.

- **TIK** (128 bits) = KDF (K, RandIK).

- **EK** (128 bits) = KDF (K, RandEK).

- **RandUESecCapab** (6 bits) : Un nombre aléatoire généré par l'UE dans le but de permettre au MME et à l'utilisateur de se mettre d'accord et en toute sécurité sur les algorithmes de chiffrement et d'intégrité (Chosen UEsecCap).

- **UESecCapab** (12 bits): la liste des algorithmes de chiffrement et d'intégrité supportés par l'UE.

- **(M), MACx** (32 bits): Le code Mac d'intégrité du message (M), généré en utilisant la clé d'intégrité x.

- **PKH**, **PKM**: Les clés publiques du réseau d'origine HSS, et du réseau de service MME respectivement.

- **KRM**, **KRH**: dénote la clé privée de MME et de HSS respectivement.

- **RandMH1** (128 bits) : valeur aléatoire générée par le MME et qui sert à la génération d'une clé de sécurité.

- **RandHM2** (128 bits) : valeur aléatoire générée par le HSS et qui sert à la génération d'une clé de sécurité.

- **MHK** (256 bits) : clé secrète partagée entre le MME et le HSS et donnée par MHK=KDF (RandMH1||RandHM2, GUMMEI, ID_{HSS}). Elle est divisée en deux clés MHIK et MHEK.

- **MHIK**, **MHEK** (128 bits) : sont les clés d'intégrité et de chiffrement respectivement, extraites du MHK telle que : MHIK est la première moitié de la clé MHK (les 128 bits de poids le plus fort) et MHEK est la deuxième moitié. Ces deux clés sont destinées à assurer l'intégrité et le chiffrement des messages échangés entre MME-HSS.

- **Session-id** (16 bits) : numéro utilisé à chaque demande des données d'authentification ou interaction entre le MME et le HSS. Il aide à indiquer à quel utilisateur ou à quelle session les messages échangés appartiennent.

- **NSQN** (32 bits) : Nombre généré par le HSS et envoyé chiffré à l'UE. Il servira à éviter les attaques par rejoue et DoS sur le HSS/AuC.

- **UMIK**, **UMEK** (128 bits): ce sont deux clés partagées entre l'UE et le MME pour authentifier et chiffrer, respectivement les messages AKA échangés entre eux. On les utilise jusqu'à le début d'utilisation des clés NAS d'EPS-AKA, et elles sont données par KDF (EK, Distingueur d'Alogrithme, Alg.ID)

- $\{m\}_K$: dénote le message *m* chiffré par la clé de chiffrement K.

4.5.2 Lancement du protocole FP-AKA

L'avantage majeur du protocole FP-AKA consiste à assurer tout le temps et entre les différentes entités communicantes, un canal sécurisé, et protégé en intégrité et en confidentialité. Ceci vise à identifier l'émetteur et échanger les données d'authentification en toute sécurité. Les premiers messages sont protégés en se basant sur le chiffrement asymétrique puisqu'il n'y a pas encore des clés symétriques établies. Via ces messages on échange certains paramètres afin d'établir des clés symétriques communes. Les messages échangés après, seront protégés en utilisant le chiffrement symétrique, puisque ce genre de chiffrement est beaucoup plus rapide que les algorithmes asymétriques. Pour pouvoir appliquer le FP-AKA, chaque abonné doit posséder un seul certificat électronique contenant la clé publique de son réseau d'origine (PKH). Chaque MME doit encore posséder un certificat électronique contenant la clé publique du HSS auquel il est attaché avec son réseau de service. Le MME doit aussi avoir les certificats des clés publiques des HSS avec lesquels l'opérateur local a fait un accord mutuel d'itinérance. Le HSS à son tour sauvegarde les certificats des MME de son réseau de service et les certificats des MME d'autres réseaux de services avec lequel son opérateur a fait un accord. Tous les certificats sont délivrés et obtenus à partir d'une autorité de confiance CA (certification Authority) comme le Verisign ou Thawte. Les détails du processus d'acquisition du certificat numérique se trouvent dans [Farrell, 2000].

La figure 4.17 montre la série des messages de signalisation échangés durant l'application du protocole FP-AKA. Expliquons maintenant les différents messages transmis, de 1 à 9, chacun à part.

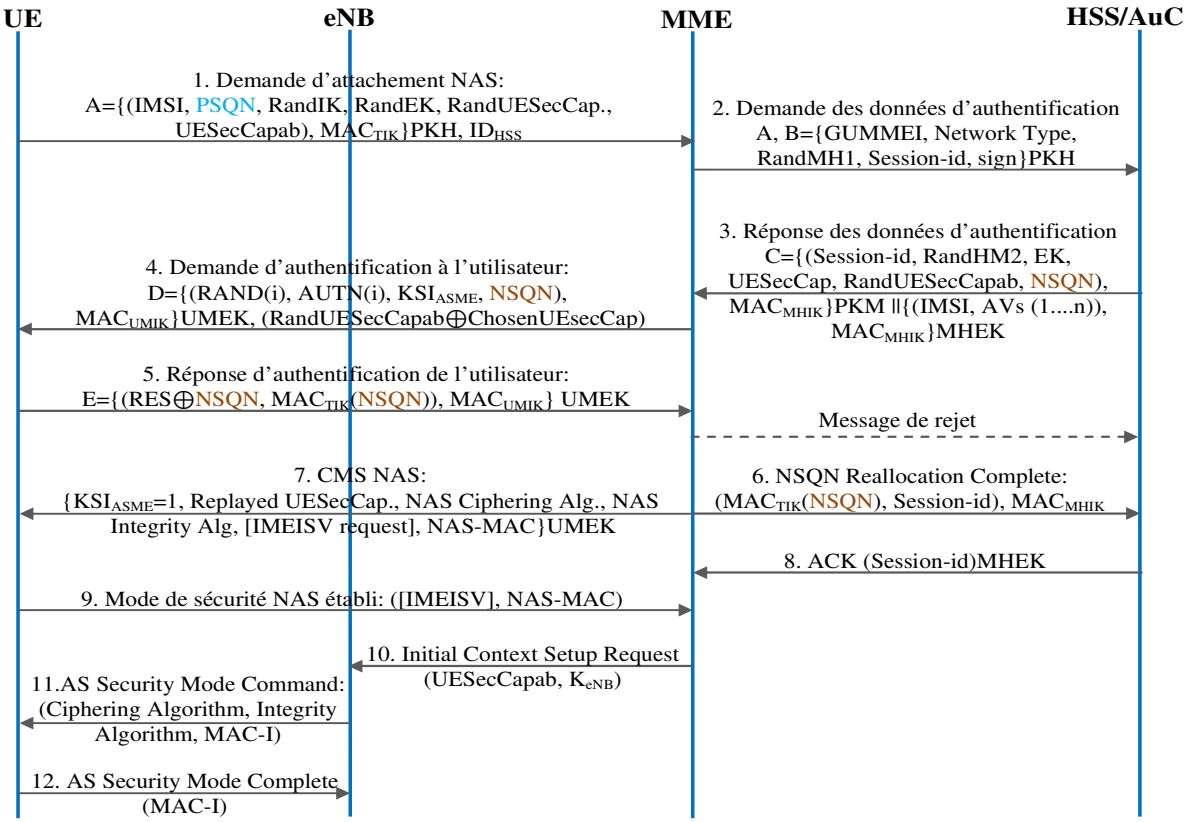

Figure 4.17. Procédure de signalisation de FP-AKA

1) Message émis de *l'UE* à *MME : Demande d'attachement NAS* A= {(IMSI, PSQN, RandIK, RandEK, RandUESecCap, UESecCapab), MAC$_{TIK}$}PKH, ID$_{HSS}$

L'UE commence par générer les trois nombres aléatoires RandIK, RandEK, et RandUESec Capabilities. La première fois où l'UE fait une demande d'attachement, il génère une valeur supplémentaire PSQN (Previous SQN). Pour ses accès au réseau après, l'UE utilisera une valeur spécifique de PSQN égale à NSQN générée par le HSS et délivrée à l'UE dans les messages de signalisation. Les 4 valeurs aléatoires générées seront transmises dans le message de demande d'attachement avec l'IMSI et les capacités de sécurité d'UE (UESecCapab). Le message est protégé en intégrité en utilisant la clé d'intégrité TIK et la fonction de sécurité f12 (utilisée par [Caragata *et al.*, 2011-a]), où TIK= KDF (K, RandIK). Le message est chiffré en utilisant la clé publique PKH, du réseau d'origine, stockée dans l'UICC. Après le chiffrement, le message obtenu A est ajouté à l'identifiant du HSS, {A, ID$_{HSS}$}, et ils sont envoyés tous les deux dans la demande d'attachement NAS.

2) Message émis de *MME* à *HSS : Demande des données d'authentification:* A, B= {GUMMEI, Network Type, RandMH1, Session-id, sign} PKH

Après la réception de la demande d'attachement de l'abonné, le MME extrait l'ID$_{HSS}$ afin de savoir vers quel réseau d'origine HSS il doit émettre sa requête et quelle clé publique PKH

utilise-t-il pour la chiffrer. Le MME crée un message contenant: son identité GUMMEI, le type du réseau mobile (LTE ou UMTS), une valeur aléatoire RandMH1, et un identificateur de session Session-id. La RandMH1 sera utilisée dans le processus de génération d'une clé commune MHK entre MME et HSS. Le message est signé par la clé KRM afin de protéger son intégrité et authentifier le MME émetteur, et il est chiffré ensuite par la clé PKH pour former le message B. Ce dernier est transmis avec A au HSS convenable.

3) Message émis de *HSS* à *MME : Réponse des données d'authentification:*

C= {(Session-id, RandHM2, EK, UESecCap, RandUESecCapab, NSQN), MAC_{MHIK}}PKM || {(IMSI, AVs (1....n)), MAC_{MHIK}}MHEK

C= $\{C1\}_{PKM}$ ||$\{C2\}_{MHEK}$

Le HSS doit déchiffrer le message reçu formé de deux parties A et B qui sont chiffrées toutes les deux par sa clé publique PKH. La première, A, est envoyée de l'UE et la deuxième du MME. Ensuite, le HSS prépare sa réponse pour MME, formée aussi de deux parties C1 et C2. Les deux sont destinées au MME, mais la première C1 est chiffrée avec la clé publique PKM du MME, et la deuxième C2 qui contient beaucoup d'informations (n vecteurs AV et autres) est chiffrée avec la clé symétrique MHEK et non pas avec PKM pour la rapidité comme le chiffrement symétrique est beaucoup plus rapide que l'asymétrique. Jusqu'au moment la clé MHEK n'est pas encore établie. Le HSS déchiffre tout d'abord A et ensuite, en déchiffrant B il génère la clé maîtresse MHK qui sera divisée en deux autres clés pour assurer la sécurité entre HSS et MME. Ces deux clés sont : MHEK pour le chiffrement des messages et MHIK pour leur intégrité. Expliquons maintenant qu'est ce que le HSS effectue lors la réception et le déchiffrement de A et B, pour présenter après les deux parties émises C1 et C2.

La procédure de déchiffrement de A est donnée par l'organigramme (a) de la figure 4.18. Le déchiffrement se fait à l'aide de la clé privée de HSS appelée KRH. Le HSS vérifie l'IMSI et sa PSQN correspondant stockés tous les deux dans l'enregistrement de l'utilisateur. Si l'IMSI et le PSQN sont correctes, la clé permanente K est extraite afin de continuer le fonctionnement normal. Sinon le HSS suppose que c'est une attaque et il jette la demande. Notons que seulement dans la première *demande d'attachement*, le HSS ne dispose d'aucune valeur PSQN stockée pour l'abonné et il accepte dans ce cas la valeur spécifiée par l'UE. Une fois la clé K est extraite, elle sera utilisée avec le RandIK reçu pour dériver la clé d'intégrité TIK. Ceci est faite, exactement comme l'UE a fait, avec une fonction KDF connue des deux côtés. Ensuite l'intégrité de A est vérifiée : en cas d'échec le HSS rejette la demande, sinon le HSS génère la clé de chiffrement EK = KDF (K, RandEK) où RandEK est la valeur reçue dans A. On va arriver à un moment où tous les abonnés du réseau qui ont fait déjà le premier attachement au réseau (ont leurs mobiles sous tension) disposent d'un numéro PSQN (stocké dans l'UICC et dans HSS). Dans ce cas l'étape de comparaison du PSQN reçu (voir figure 4.18 (a)) à celui stocké dans HSS est toujours obligatoire.

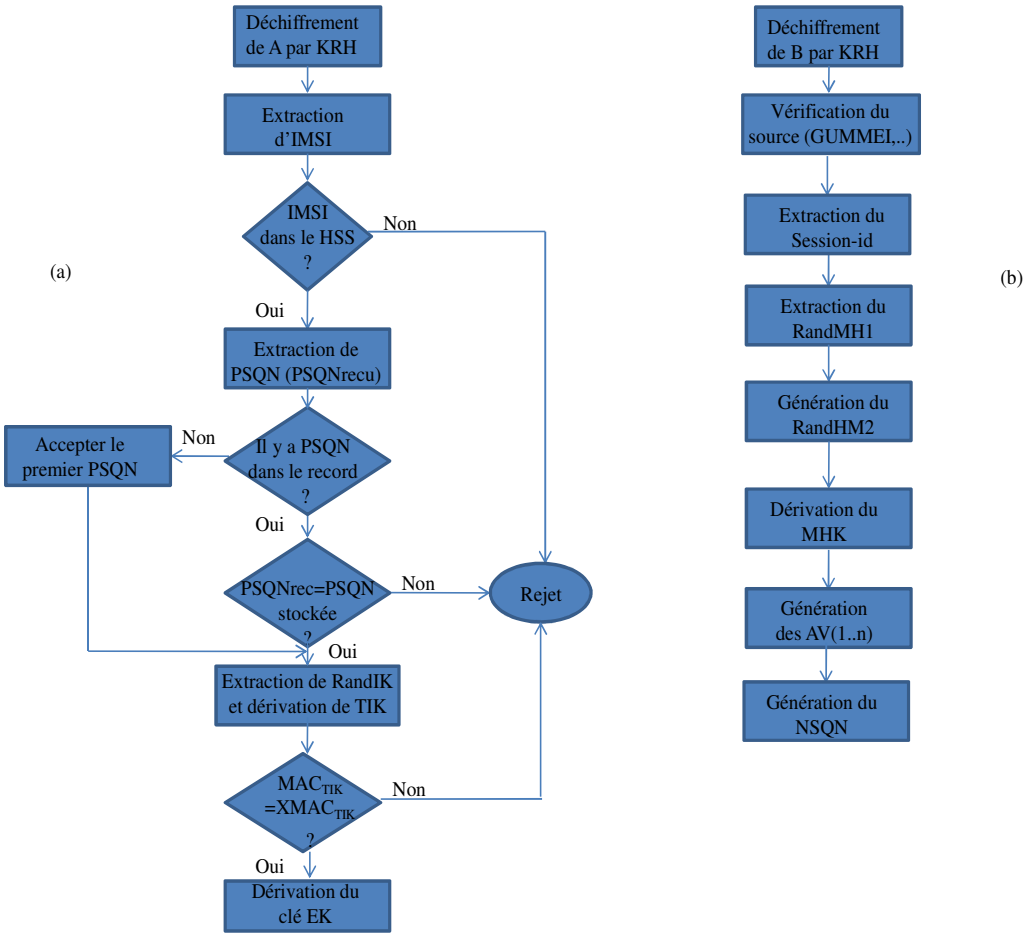

Figure 4.18. Comportement de HSS/AuC à la réception du message 2 (a) déchiffrement de A et protection contre les attaques (b) déchiffrement de B

Le HSS termine le déchiffrement de A et puis il commence à déchiffrer l'autre partie reçue B représentée par l'organigramme (b) de la figure 4.18. Ceci se fait par sa clé privée PKH.

Il commence par vérifier si le SN id, et qui fait parti du GUMMEI reçu, est valide et correspond à l'un des réseaux de services autorisés et enregistrés dans sa base de données. Si les identités de MME et celle du réseau de service, qui forment le GUMMEI, ont été vérifiées, ainsi que l'intégrité du message reçu B, le HSS sauvegarde le session-id et génère la clé MHK. Cette clé est générée en utilisant le RandMH1 reçu et une valeur secrète aléatoire générée par lui-même RandHM2. MHK constitue une clé secrète maîtresse entre le MME et le HSS et qui sera divisée en deux clés MHEK et MHIK pour assurer respectivement le chiffrement symétrique et l'intégrité des messages transmis.

Lorsque le HSS termine le déchiffrement et la vérification de l'intégrité de A et B, il génère (avec son AuC) et comme dans l'EPS-AKA, n vecteurs d'authentification AV (1 ... n) qui constituent la composante la plus importante dans le message *réponse des données d'authentification* émis par le HSS au MME. Ce message que nous avons appelé C est formé par $\{C1\}_{PKM} \| \{C2\}_{MHEK}$. La première partie C1 comprend : l'identifiant de la session (reçu dans B) pour informer le MME à

quelle session cette réponse appartient, le RandHM2 pour permettre au MME de dériver la même MHK, l'EK qui sera utilisé comme une clé mère entre l'UE et le MME, et un nouveau numéro de séquence NSQN pour le livrer à l'utilisateur convenable (ayant l'IMSI dans C2). Le HSS protège l'intégrité de tous les champs inclus dans C1 par le code MAC généré à l'aide de la clé MHIK. Ce code MAC_{MHIK} est concaténé avec les données protégées pour former C1. Ce dernier est chiffré à l'aide de la clé publique de MME (PKM).

La deuxième partie du message de réponse C2 contient l'identité de l'abonné IMSI et les vecteurs AV (1 ... n) correspondants, protégés en intégrité par MHIK et chiffrés symétriquement par MHEK. Les deux parties $\{C1\}_{PKM}$ et $\{C2\}_{MHEK}$ bien protégées, sont envoyées au MME comme la *réponse des données d'authentification* .

4) Message émis de *MME* à *l'UE : Demande d'authentification à l'utilisateur:* {(Rand(i), AUTN(i), KSI_{ASME}, NSQN), MAC_{UMIK}}UMEK, (RandUESecCapab ⊕ ChosenUEsec Cap)

En recevant le message numéro 3, C, le MME commence à déchiffrer C1 avec sa clé privée KRM. Aucun autre MME ne peut déchiffrer C1. Il vérifie ensuite le session-id reçu afin d'authentifier le HSS puisque c'est la seule entité qui peut avoir cette valeur (contenu dans le message B transmis avant). Pour générer la clé MHK et avoir les deux clés MHIK et MHEK, le MME extrait le RandHM2, le concatène avec son RandMH1 et calcule MHK comme étant égale à KDF (RandMH1||RandHM2, GUMMEI, ID_{HSS}). L'intégrité de C1 est vérifiée à l'aide de MHIK. La deuxième partie C2 de C, est déchiffrée en utilisant MHEK, et son intégrité est vérifiée à l'aide de MHIK. En cas d'échec un code d'erreur sera envoyé à HSS. Le succès de ces vérifications affirme à MME qu'il partage avec le HSS les mêmes clés symétriques MHIK et MHEK. À partir de ce moment tous les futurs messages échangés entre MME et HSS seront sécurisés par ces clés symétriques. La clé EK contenue dans C1 est utilisée par le MME pour dériver deux autres clés UMIK et UMEK pour assurer la protection de l'intégrité et la confidentialité des messages transmis durant le protocole AKA entre l'UE et le MME. Ce dernier choisit parmi la liste émise par l'utilisateur dans UESecCapab, les algorithmes de chiffrement et d'intégrité à utiliser. Les identités des algorithmes (Algorithm-ID) choisis (ChosenUEsecCap) seront masquées, via une opération XOR, par la valeur aléatoire RandUESecCapab.

Avec la clé MHEK, le MME déchiffre C2 et choisit parmi les AV (1, ..., n) reçus, un vecteur d'authentification AV (i) pour en extraire RAND(i), et AUTN(i). Ces paramètres seront ajoutés à KSI_{ASME} et à NSQN pour les protéger tous en intégrité avec la clé UMIK et en calculant le code MAC_{UMIK} ajouté aux paramètres. Pour plus de sécurité, toutes ces valeurs sont chiffrées à l'aide de l'algorithme de chiffrement choisi et la clé UMEK. Les données chiffrées seront concaténées avec la combinaison, via l'opération XOR, des deux valeurs RandUESecCapab et ChosenUEsecCap. Le message formé D, est transmis à l'UE en tant que la *demande d'authentification à l'utilisateur*.

5) Message émis de *l'UE* à *MME : Réponse d'authentification d'utilisateur:* {(RES⊕NSQN, MAC_{TIK}(NSQN)), MAC_{UMIK}} UMEK

À la réception du message numéro 4, D, l'UE utilise sa valeur aléatoire RandUESecCap pour démasquer les algorithmes sélectionnés par le MME comme suivant: RandUESecCapab ⊕ (RandUESecCapab ⊕ ChosenUEsecCap). Le résultat indique les algorithmes de chiffrement et d'intégrité qui doivent être utilisés par l'UE. Ce dernier dérive les clés symétriques UMIK et UMEK à l'aide de la clé EK et les codes des algorithmes (Algorithm-ID) reçus (processus similaire à la dérivation de K_{NASint} et K_{NASenc}). L'UE peut maintenant déchiffrer le message D, et vérifier son intégrité en utilisant les algorithmes et les clés convenables. À partir de ce message, l'UE extrait la valeur de NSQN et il la sauvegarde afin de l'utiliser lors de la prochaine AKA. À partir des paramètres RAND(i), AUTN(i), et KSI_{ASME} il vérifie l'authenticité du réseau de service. En cas de succès : RES sera créé, une copie de NSQN sera retournée au MME sous la forme (RES ⊕ NSQN), et un MAC sera généré pour ce NSQN à l'aide de la clé TIK. Le dernier paramètre MAC_{TIK}(NSQN) sera transmis après à HSS. L'UE envoie le message de réponse E contenant (RES ⊕ NSQN) et le MAC_{TIK}(NSQN), protégé tous les deux en intégrité et chiffrés en utilisant les mêmes clés et les mêmes algorithmes utilisés pour l'envoie du message précédent par le MME.

6) Message émis de *MME* à *HSS : NSQN Reallocation Complete*: (MAC_{TIK}(NSQN), Session-id), MAC_{MHIK}

Après le déchiffrement et la vérification d'intégrité du message numéro 5, *réponse d'authentification de l'utilisateur*, le MME extrait RES en utilisant le NSQN (gardé lors de l'émission du message précédent numéro 4) comme suivant: NSQN⊕ (RES⊕NSQN). Si le RES extrait, est identique au RES attendu (XRES), le réseau authentifie l'UE et s'assure en même temps que le NSQN a été bien reçu par l'UE. Dans ce cas, et pour notifier le HSS que l'utilisateur est devenu en possession de NSQN, le MME envoie le message 6 contenant le code MAC_{TIK}(NSQN) (qu'il a reçu dans le message précédent E) accompagné du session-id correspondant, et protégés tous les deux par la clé MHIK. Le HSS sauvegarde dans ce cas la valeur NSQN qu'il a généré lui-même au début (après la réception du message 2) pour cet utilisateur. Si RES≠XRES, le MME informe le HSS de cet échec en lui envoyant un message de rejet (*NSQN Reallocation Reject*) contenant l'identificateur de la session (session-id) et protégé par MHIK et MHEK. Dans ce cas, le HSS détruit les différentes valeurs associées à cette session et le standard recommande la transmission d'un message de rejet (*Authentication Reject message*) par le MME à l'UE.

7) Message émis de *MME* à *l'UE : Commande du mode de sécurité NAS, CMS NAS:* {KSI_{ASME}=1, Replayed UESecCap., NAS Ciphering Alg., NAS Integrity Alg, [IMEISV request], NAS-MAC}UMEK

En même temps et avec la transmission du message numéro 6, le MME envoie à l'UE le message qui annonce le début du mode de sécurité *NAS Security Mode Command* (CMS NAS). Selon le standard et d'habitude en EPS-AKA, on protège ce message seulement en intégrité avec la clé générée de l'AKA (K_{NASint}). Nous proposons maintenant de chiffrer les informations se trouvant dans ce message ainsi que leur code d'intégrité NAS-MAC, par la clé UMEK. Les algorithmes de

chiffrement et d'intégrité NAS choisis doivent appartenir à la liste des algorithmes de sécurité (UESecCapab) envoyé par l'UE.

8) Message émis de *HSS* à *MME* : *ACK* : (Session-id) MHEK

À la réception du message *NSQN Reallocation Complete*, le HSS vérifie l'intégrité de ce message, et vérifie si le code MAC_{TIK}(NSQN) reçu est correct et égale au MAC attendu, $XMAC_{TIK}$(NSQN). Si les deux MAC sont égaux, le HSS obtient une double confirmation: premièrement il vérifie que c'est le bon utilisateur qui a été authentifié, et il s'assure qu'il possède le bon NSQN qui le servira dans la prochaine demande d'attachement (*Attach request*) avec le HSS ; deuxièmement le HSS s'assure que les clés secrètes partagées avec le MME sont correctes et que l'entité appropriée a bien déchiffré les données transmises. Pour acquitter la mise à jour du NSQN, un message d'acquittement chiffré, contenant le session-id sera retourné au MME. Ainsi le MME s'assure que le HSS a sauvegardé le NSQN dans l'enregistrement de l'utilisateur et il lui reste d'informer l'UE de sauvegarder cette valeur.

9) UE→MME : *Mode de sécurité NAS établi*: ([IMEISV], NAS-MAC)

Après la réception du message numéro 7, l'UE déchiffre ce message par UMEK, vérifie le NAS-MAC par la clé K_{NASint} et répond au MME avec le message *Mode de sécurité NAS établi.* À partir de ce moment toute la signalisation NAS sera protégée en intégrité et chiffrée à l'aide les clés générées par la procédure EPS-AKA (K_{NASenc} et K_{NASint}). Rappelons et comme nous avons vu dans le paragraphe 3.5.4.4.2, l'un des messages NAS transmis par le MME à l'UE contient le GUTI. Nous proposons d'ajouter à ce message un indicateur (flag) pour commander l'UE à stocker le NSQN maintenu. Le message *GUTI Reallocation Complete* confirme au MME que le NSQN est stocké dans l'équipement utilisateur et que le GUTI a été bien alloué.

Les trois messages (10, 11, et 12) restants dans la figure 4.17, transmis de MME à l'eNB et échangés entre l'UE et l'eNB, sont les mêmes comme en EPS-AKA expliqué dans le chapitre 3.

À rappeler que cette procédure est déclenchée seulement : lors de la première connexion RRC lorsqu'un utilisateur met son portable en marche, et lorsqu'un réseau de service légal ou une fausse eNB demande à l'UE d'envoyer son identité permanente. Après le premier attachement et après l'établissement de la clé MHK entre MME et HSS (dans la première session), tous les messages de signalisation échangés, entre ces deux entités seront protégés par cette clé partagée MHK. Ceci vise à réduire le coût et le délai de calcul en comptant sur le chiffrement symétrique et en évitant l'asymétrique qui coûte cher. Pour assurer une sécurité parfaite, on doit changer la clé MHK après une certain période ou après un certain nombre de sessions établies entre MME et HSS. Le nombre de sessions est indiqué par l'opérateur suivant sa politique de sécurité appliquée.

Donc l'application du protocole FP-AKA, après le premier accès, reste la même avec les mêmes messages échangés avec un petit changement qui affecte seulement les messages B et C (voir figure 4.19). Ils ne sont plus chiffrés par les clés publiques PKH et PKM, mais ils sont protégés par les clés MHIK et MHEK.

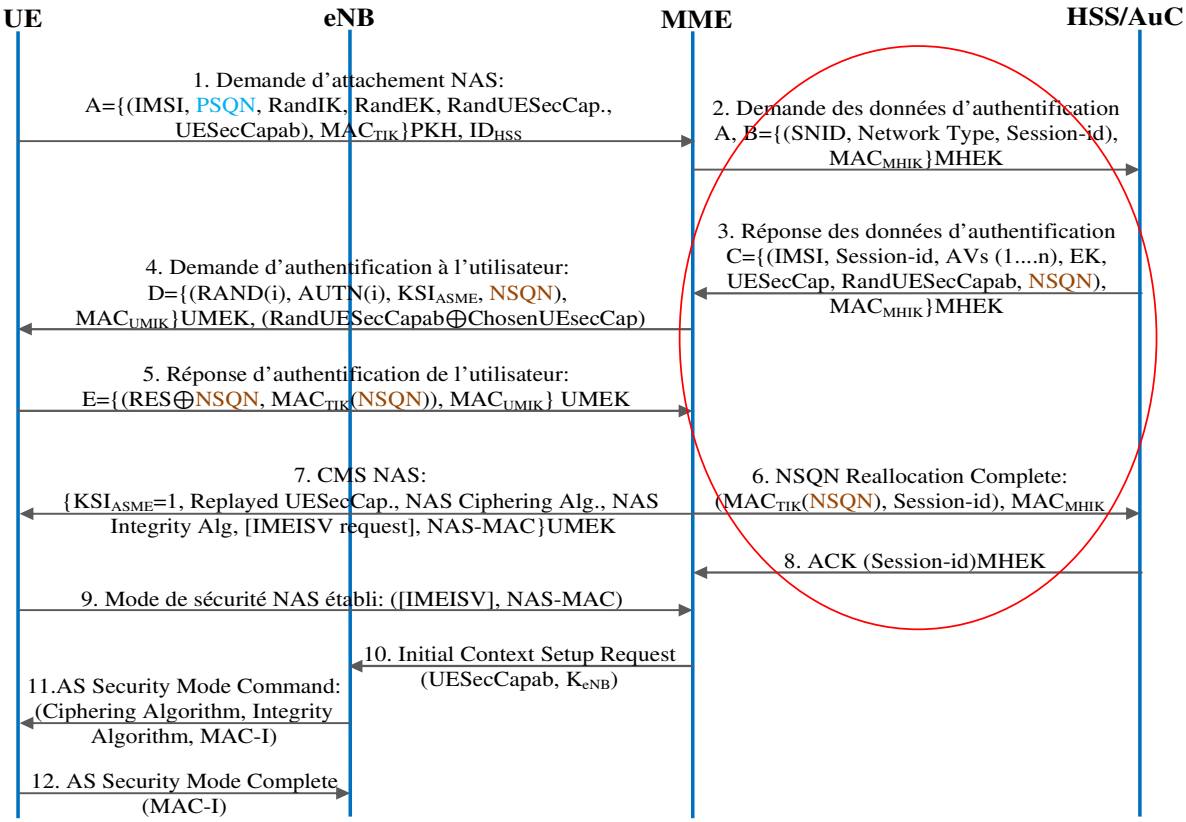

Figure 4.19. Protocole FP-AKA lancé à partir de la deuxième session

4.5.3 Composantes de sécurité du protocole FP-AKA

L'application de notre protocole FP-AKA nécessite l'utilisation de plusieurs fonctions de sécurité : Une fonction pour la génération des nombres secrets aléatoires, une fonction pour la dérivation des clés KDF, deux fonctions représentant les algorithmes de chiffrement et d'intégrité, et un algorithme de chiffrement asymétrique.

4.5.3.1 Génération des nouvelles clés : fonctions et paramètres d'entrée

Nous proposons d'utiliser la fonction f0 pour la génération des différentes valeurs aléatoires ***RandIK, RandEK, RandUESecCap, RandMH1 et RandHM2***, ou bien un générateur RBG (Random Bit Generator) approuvé [Barker et Kelsey, 2012]. La méthode de génération de NSQN peut être laissée à l'opérateur. Il peut utiliser par exemple un des mécanismes utilisés pour la gestion des numéros de séquence SQN$_{HE}$ et qui est indiqué dans [TS 33.102, 2012].

Nous avons vu dans le chapitre précédent (paragraphe 3.5.4.3.2) que chacune des entités : l'UE, le MME et le HSS, dispose d'une fonction de dérivation de clés KDF (*Kin, S*), basée sur l'algorithme HMAC-SHA-256. Dans notre protocole, nous avons cinq clés à dériver : les clés **TIK**, **EK** (entre UE-HSS), **MHK** (entre MME-HSS), **UMIK** et **UMEK** (entre UE-MME). Pour dériver ces clés, nous proposons d'utiliser la même fonction KDF=HMAC-SHA-256 approuvée et implémentée déjà dans les différentes entités. Ce qui varie dans la dérivation d'une clé à

l'autre, ce sont les entrées (*Kin* et *S*) de cette fonction. Le tableau 4.3 présente les différents paramètres d'entrée que nous proposons affecter, pour la génération de chaque clé du FP-AKA.

Clé(s) dérivé(s) à la sortie	**Clé secrète d'entrée *Kin***	***S*=FC||P0||L0||…..||P*n*||L*n***			**Longueur de la clé dérivée (bits)**
		FC alloué	P0, P1, …, P*n*	L0, L1,.., L*n*	
TIK	K	0x15	RandIK	0x0010,	128
UMEK, UMIK	EK	0x15	Distingueur de type d'algorithme, Identifiant de l'algorithme	0x0001, 0x0001	128
EK	K	0x1D	RandEK	0x0010	128
MHK (MHIK, MHEK)	RandMH1||RandHM2	0x1E	GUMMEI, ID_{HSS}	0x0006, 0x0004	256 (128, 128)

Tableau 4.3. Entrées *Kin* et *S* de la fonction de dérivation des clés dans FP-AKA

Comme les trois clés **TIK**, **UMEK** et **UMIK** sont des clés feuilles, utilisées avec des algorithmes d'intégrité ou de chiffrement, alors le paramètre d'entrée FC doit être égale à 0x15 d'après [TS 33.401, 2012] pour ces trois clés. La dérivation des clés UMEK et UMIK nécessite l'utilisation de l'entrée supplémentaire 'distingueur du type d'algorithme' afin de distinguer entre ces deux nouvelles clés feuilles et les autres clés feuilles définies (comme K_{NASenc}, K_{NASint},…). Donc pour ces deux clés, il faut définir deux nouvelles valeurs pour ce champ. D'après [TS 33.401, 2012], les valeurs réservées pour une future utilisation sont les valeurs comprises entre 0x07 et 0xF0. Pour cela nous proposons d'allouer les valeurs 0x07 et 0x08 pour dériver les clés UMEK et UMIK. Ces nouvelles valeurs, présentées dans le tableau 4.4, indiquent dorénavant que ces clés servent pour le chiffrement et pour l'intégrité des messages d'authentification AKA et le chiffrement du CMS NAS.

Distingueur d'algorithme	**Valeur**
Alg-chiff-NAS	0x01
Alg-int-NAS	0x02
Alg-chiff-RRC	0x03
Alg-int-RRC	0x04
Alg-chiff-UP	0x05
Alg-int-UP	0x06
Alg-chiff-AKA	**0x07**
Alg-int-AKA	**0x08**

Tableau 4.4 : Les anciennes et les nouvelles valeurs proposées du 'distingueur d'algorithme'

Les clés **EK** et **MHK** ne sont pas des clés feuilles, et nous proposons de les servir pour des nouvelles fins de sécurité non spécifiées dans les standards de l'EPS. Pour cela il est également nécessaire de spécifier de nouveaux codes FC pour la dérivation de chacune de ces clés. D'après le paragraphe 3.5.4.3.2, il y a trois valeurs disponibles de FC pour des futures utilisations. Nous proposons d'allouer la valeur 0x1D comme code FC pour la fonction de dérivation de la clé EK, et la valeur FC= 0x1E comme code d'entrée pour la dérivation de la clé MHK. Cette dernière est dérivée lors de la première demande d'attachement à partir de deux secrets aléatoires, RandMH1||RandHM2, et prend comme entrées supplémentaires *S* les identifiants GUMMEI et ID_{HSS} (GUMMEI est de 6 octets et l'ID_{HSS} de 28 bits). La sortie de cette fonction n'est que la clé MHK qui donnera naissance à deux clés MHK=MHIK||MHEK.

4.5.3.2 Algorithmes et fonctions de sécurité dans FP-AKA

Pour la génération du code MAC_{TIK} dans les messages A et E du protocole FP-AKA, nous proposons d'utiliser la fonction f12 basée sur l'algorithme HMAC-SHA-256. Cette fonction a besoin de trois paramètres à son entrée [Barker *et al.*, 2009]: La clé d'intégrité TIK, la longueur en octets du code MAC (champ appelé MACLen égale à 4 dans notre cas), et le message lui-même à protéger MACData.

Les messages AKA entre l'UE et le MME sont protégés en intégrité et en confidentialité par les fonctions f8 et f9. La protection des données (en chiffrement et intégrité) entre MME et HSS nécessite l'implémentation des algorithmes robustes f8, f9 au niveau du HSS. Ainsi tous les messages seront protégés comme le standard en intégrité et contre la répétition ainsi que le chiffrement.

Le chiffrement asymétrique des messages A, B et C1 peut se faire par l'algorithme RSA. Nous proposons d'utiliser la variante sûre de RSA qui est le RSA-OAEP (RSA- Optimal Asymmetric Encryption Padding) [Fujisaki *et al.*, 2004] décrite dans l'annexe B3.

4.5.4 Analyse de la robustesse du protocole FP-AKA

Les messages échangés durant le protocole FP-AKA sont tous protégés en confidentialité et en intégrité. Ce qui fait que la confidentialité de l'utilisateur, et la protection contre les différentes attaques sont garanties. Les nouvelles améliorations proposées dans le FP-AKA, répondent à la philosophie de la sécurité recommandée pour l'EPS et respectent toutes les exigences imposées par le groupe de travail de sécurité de 3GPP. Appliquer le FP-AKA améliore nettement la sécurité du réseau EPS en assurant une protection complète contre toutes les attaques déjà mentionnées:

- ✓ Le chiffrement de l'IMSI par la clé publique du HSS en introduisant des valeurs aléatoires et en chiffrant par RSA-OAEP assure la **confidentialité de l'IMSI** (et par suite de l'utilisateur) et évite les menaces d'identification et de traçage des utilisateurs. Le chiffrement par la PKH garantit que personne (non désirée) ne peut récupérer l'IMSI ou les informations incluent dans la *demande d'attachement*, à partir du texte chiffré A (sans avoir la clé privée).

- ✓ La protection de l'intégrité du message A contenant les capacités de sécurité de l'UE par le code MAC_{TIK}, assure la protection contre le « ***bidding down attack*** » et par suite contre l'attaque **DoS sur l'UE**.

- ✓ Le champ NSQN est conçu pour éviter **l'attaque par rejoue** et **l'attaque DoS sur le HSS/AuC**. En effet, comme le HSS affecte à l'UE une nouvelle valeur NSQN pour chaque accès, ceci garantit une sécurité contre les attaques de rejoue. Surtout que ce paramètre est échangé entre les nœuds en toute sécurité, chiffré et protégé en intégrité. Comme l'attaque par rejoue peut se transformer en la répétant avec une fréquence très élevée à une attaque DoS, par suite le NSQN protège aussi contre cette dernière attaque (DoS).

- ✓ L'envoie de l'IMSI avec les vecteurs d'authentifications et la transmission de la session-id dans le même message C (réponse d'authentification), permettent d'éviter **l'attaque sur les réponses des données d'authentification** et la violation des propriétés de l'authentification.

- ✓ Le changement permanent de la clé *MHK* (partagée entre MME et HSS), périodiquement ou après son utilisation pour une certaine quantité de bits, garantit une excellente sécurité entre les nœuds MME et HSS. Lorsqu'on désire changer cette clé, chacun des deux nœuds génèrent une valeur aléatoire, RandMH1 et RandMH2, et la transmet à l'autre nœud afin de calculer une nouvelle valeur de MHK comme étant égale à KDF (RandMH1‖RandHM2, GUMMEI, ID_{HSS}). Ce changement de la clé en permanence permet de respecter les recommandations à ce propos.

- ✓ Le compromis de la clé privée du MME ne permet pas de compromettre une clé MHK précédente et par suite on ne pourrait déchiffrer aucun message échangé avant, entre MME et HSS. De même, avec la possession de cette clé privée, on ne peut pas connaître ni les vecteurs d'authentification ni l'IMSI des utilisateurs. Même les clés UMEK et UMIK utilisées entre l'UE et le MME, restent sécurisées dans ce cas et on ne peut pas les dévoiler.

- ✓ La protection, en intégrité et en confidentialité, de la liaison entre MME et HSS empêche **les attaques de type MITM** (compromis des AV et blocage des services).

- ✓ Le chiffrement et l'intégrité des messages AKA échangés entre l'UE et le MME évitent l'**attaque contre la clé secrète permanente K** (attaque à texte en clair connu et à texte chiffré seulement) ainsi que l'attaque **DoS contre l'UE**.

- ✓ Le champ $MAC_{TIK}(NSQN)$ reçu par le HSS dans le message 6, lui permet de s'assurer que le FP-AKA s'est bien déroulé en succès. Ce champ garantit aussi au HSS que l'utilisateur possède le bon NSQN, et que le MME partage avec lui les bonnes clés secrètes MHIK et MHEK.

- ✓ Lorsque l'UE découvre qu'il y a un problème dans le message 7 (CMS NAS), il envoie au MME un message de rejet (*NAS Security Mode Reject*) bien protégé avec les clés UMIK et UMEK. Cette protection permet d'éviter l'attaque **DoS contre l'UE**.

Notre protocole assure donc une protection complète contre les différents types d'attaques. Il ne présente aucune vulnérabilité. Ca prouve qu'il est un vrai candidat pour remplacer le protocole standard de 3GPP EPS-AKA.

4.6. Analyse de la qualité de Service des protocoles AKA étudiés

Dans cette section, nous allons analyser et comparer les quatre protocoles EPS-AKA, SE-AKA, EC-AKA et FP-AKA selon différents facteurs afin d'estimer la performance et la qualité de service QoS (Quality of Service) de chacun de ces protocoles.

La sécurité et la performance ne sont pas nécessairement opposées. En général, un niveau de sécurité élevé a un coût de traitement élevé et des grands taux de données ajoutées (additional overhead). La performance d'un protocole est estimée selon ses caractéristiques en comparaison avec les besoins de l'application. Habituellement, il n'y a pas protocole meilleur que les autres, puisque la prise de décision est fondée sur les besoins de l'application, et sur la politique de l'opérateur suivi.

Pour évaluer la performance de chaque protocole, nous allons considérer les paramètres suivants:

- Sécurité/Risque: la sécurité d'un protocole est définit comme sa capacité à résister aux attaques, et le risque est la probabilité qu'une attaque puisse réussir à violer le protocole. Plus le coût et les efforts pour exploiter une certaine vulnérabilité sont élevés, plus la probabilité que l'attaque réussit diminue.
- Coût: C'est les dépenses d'investissement (CAPEX-Capital Expenditure) et les dépenses de fonctionnement/d'exploitation (OPEX-Operational Expenditure) qui coûtent l'application d'un certain protocole.
- Taux de données ajoutées (overhead): C'est le trafic ajouté sur les interfaces de transmission afin de pouvoir appliquer le protocole considéré.
- Délai: C'est le retard global résultant de l'application d'un protocole. Un grand délai causera parfois des problèmes pour certains services. Il conduira par exemple à un taux faible de réussite d'appel en cas de handover.

Etudions maintenant chaque paramètre et estimons-le pour chacun des protocoles étudiés. Nous ferons ainsi un classement des protocoles pour chaque paramètre à part pour déduire enfin leur performance et leur valeur vis-à-vis de la qualité de service QoS.

4.6.1 Sécurité/Risque

Les deux termes sécurité et risque sont inversement proportionnel l'un à l'autre. Plus le protocole est sécurisé, plus le risque est faible.

Le paramètre 'risque' est défini comme suivant :

$$\textbf{Risque = Valeur d'actif * menace perçue * vulnérabilité} \qquad (4.1)$$

Avec : la 'valeur d'actif' est la valeur et l'importance que le protocole joue dans le réseau EPS, et le deuxième paramètre 'menace perçue' est la valeur des menaces que le protocole peut subir, et le dernier paramètre 'vulnérabilité' reflète les faiblesses du protocole.

Les deux premiers paramètres 'valeur d'actif' et 'menace perçue' sont les mêmes pour les quatre protocoles étudiés. Alors pour évaluer les protocoles vis-à-vis du paramètre 'risque' il suffit donc d'évaluer seulement le paramètre 'vulnérabilité' pour chaque protocole. Ce paramètre est la facilité d'exploiter une faiblesse du protocole ou le nombre d'attaques possibles.

Nous avons déjà étudié la capacité de chaque protocole à résister contre tous les types d'attaques possibles. Nous avons montré la vulnérabilité de chaque protocole en dévoilant ses points forts et ses points faibles. Le tableau 4.5 résume toute cette étude qui montre chaque protocole s'il résiste ou non contre chaque type d'attaque.

Vulnérabilité	EPS-AKA	SE-AKA	EC-AKA	**FP-AKA**
1-Assurer la Confidentialité de l'IMSI	Non	Oui (cassé)	Oui	Oui
2-Résistance contre l'attaque par rejoue	Non	Non	Non	Oui
3- Résistance contre l'attaque DoS de l'UE	Non	Non	Oui	Oui
4- Résistance contre le blocage des services par un MITM	Non	Non	Oui	Oui
5-Confidentialité de l'interface MME- HSS	Non	Oui	Oui	Oui
6- Résistance contre les attaques sur les réponses des données d'authentification	Non	Oui	Non	Oui
7- Résistance contre l'attaque DoS de HSS	Non	Non	Non	Oui
8-Résistance contre l'usurpation d'identité de MME	Non	Non	Non	Oui

Tableau 4.5. Comparaison de la sécurité des différents protocoles

Donc en comparant les quatre protocoles, FP-AKA est considéré comme étant le plus sécurisé ou le moins risqué vis-à-vis les attaques et les vulnérabilités. Les protocoles sont classés dans le tableau 4.6 par ordre décroissant selon la sécurité de chaque protocole.

Niveau de sécurité	Protocole AKA proposé
1	**FP-AKA**
2	EC-AKA
3	SE-AKA
4	EPS-AKA

Tableau 4.6. Liste des protocoles ordonnés selon le niveau de sécurité

4.6.2 Coût

Analysons maintenant le coût supplémentaire de chaque protocole par rapport au coût du protocole standard EPS-AKA. Dans les protocoles proposés, on n'a pas besoin d'équipements supplémentaires et ce sont les certificats électroniques des entités qui peuvent coûter de l'argent. Comme le nombre des nœuds MME et HSS n'est pas beaucoup (quelques un pas plus) leurs certificats ne coûtent pas cher. Par exemple au Liban l'opérateur 'Touch' possède seulement deux HSS et maximum deux MME, donc le coût de leurs certificats électroniques est minime. Ce qui peut coûter cher ce sont les certificats électroniques des utilisateurs si on a besoin de ca. Car le nombre des utilisateurs est énorme et peut dépasser quelques millions, donc les dépenses de fonctionnement du protocole dans ce cas sont énormes.

Les protocoles EC-AKA et FP-AKA n'ont pas besoin des dépenses supplémentaires, par rapport à l'EPS-AKA. Ces deux protocoles n'exigent ni des équipements supplémentaires ni de payer pour des certificats aux utilisateurs. Ils utilisent tout simplement le chiffrement symétrique pour envoyer leurs messages de signalisation à l'utilisateur. On peut avoir besoin simplement de mettre à jour des logiciels utiles pour pouvoir activer les protocoles. Donc pour ces deux protocoles le coût CAPEX et OPEX supplémentaire est nul.

Parmi tous les protocoles étudiés, il y a seulement le protocole SE-AKA qui compte sur des certificats électroniques des utilisateurs. Plus précisément dans ce protocole, il y a seulement le 4ème message D, (D={RAND(i), SN id, S-TMSI, $KSI_{ASME}\}_{PKU}$), envoyé du MME à l'utilisateur, qui se base sur le chiffrement asymétrique et qui utilise la clé publique de l'UE. Dans ce cas, le MME doit avoir le certificat X.509 (c'est le type de certificat le plus connu et le plus utilisé) qui contient l'identité et la clé publique de cet UE. Ce certificat doit être délivré par un tiers de confiance (par le TTP, Trusted Third Party, par exemple) appelé autorité de certification CA (Certification Authority).

Le MME doit donc avoir accès d'une manière ou d'une autre, au certificat de l'UE. Il peut accéder au certificat par l'un des trois scénarios possibles :

1. Le certificat de l'UE est stocké dans le HSS de son réseau d'origine;

2. Chaque réseau possède sa propre base de données contenant tous les certificats de ses utilisateurs;

3. Le certificat de l'UE est disponible dans une base de données spécifique chez un tiers indépendant comme par exemple chez 'Verisign' ou 'Thawte'.

Le scénario 1 est le choix le plus convenable à cause du faible coût d'investissement (CAPEX) requis. La connexion entre les réseaux étrangers est d'habitude lente (bande passante faible) [Rohrer *et al.*, 2007], en comparaison avec la connexion entre les réseaux locaux. Ce scénario est convenable pour les demandes d'authentification provenant du même réseau, alors qu'il est moins performant pour les utilisateurs itinérants. Notons que la taille du certificat est 13 fois plus grande que la taille d'un AV émis par le HSS.

Le scénario 2 est moins cher que le scénario 3, mais les deux sont très couteux par rapport au scénario 1. En performance, ce scénario est efficace pour les connexions locales mais encore donne un rendement inférieur pour les connexions d'itinérance. Le scénario 3 est le plus coûteux. Il a des meilleurs résultats pour les utilisateurs itinérants en comparaison avec les deux scénarios 1 et 2, mais il présente de mauvaises performances pour les connexions locales. Pour le scénario 3 par exemple, et pour deux millions utilisateurs et 50$/certificat (coût approximatif actuel), le CAPEX doit coûter environ un million dollars américains par an.

Comme conclusion on peut classer les trois protocoles EC-AKA, FP-AKA et EPS-AKA avec le même niveau du coût alors que SE-KA est le plus coûteux.

4.6.3 Taux de données ajoutées sur la signalisation

Nous allons calculer dans ce paragraphe et comparer le trafic généré par les différents protocoles AKA considérés, EPS-AKA, SE-AKA, EC-AKA et FP-AKA. En améliorant la sécurité afin d'éviter les attaques, un protocole AKA transmet des données supplémentaires dans ses messages de signalisation. Pour pouvoir estimer le taux des données ajoutées sur les messages de signalisation de l'EPS-AKA, par chacun des protocoles étudiés, nous allons calculer le trafic généré en nombre de bits par tous les messages transmis durant l'application de chaque protocole. Pour cela, nous allons faire ce calcul sur les interfaces suivantes :

- Liaison montante de l'interface radio, entre l'UE et l'eNB
- Liaison descendante de l'interface radio
- Liaison montante de l'interface appelé 'backhaul', entre l'eNB et le MME
- Liaison descendante de l'interface backhaul,
- Interface de transport dans le réseau cœur, entre le MME et le HSS (dans les deux sens).

Le calcul des tailles, en bits, de chaque message de signalisation transmis sur les différentes interfaces durant chacun des protocoles AKA considérés, a été effectué. Ce calcul dépend du nombre n de vecteurs d'authentification émis du HSS au MME, et qui peut prendre n'importe quelle valeur entière (fortement conseillé entre cinq et dix). Le résultat obtenu sur chaque interface est présenté dans le tableau 4.7. L'algorithme de chiffrement asymétrique considéré, dans les protocoles où on en a besoin, est le RSA (ou le RSA-OAEP puisque les tailles calculées ne changent pas).

Protocole	Liaison radio- voie montante	Liaison Backhaul - voie montante	Liaison radio- voie descendante	Liaison Backhaul- voie descendante	Trafic moyen sur l'interface Transport (réseau cœur)
EPS-AKA	204	204	260	260	82+(n*640)
SE-AKA	1180	1180	1024	1024	2048+ ceil((n*640+8252)/1024)*1024
EC-AKA	1180	1180	394	394	3072+ ceil ((n*640+399)/1024)*1024
FP-AKA	1244	1244	330	330	1er demande d'attachement : 4284+(n*640) Durant les *i* attachements suivants: 1476+(n*640). ========== Trafic moyen égale à : 1535+(n*640).

Tableau 4.7. Trafic de signalisation, en bits, généré par les protocoles étudiés

Nous remarquons de ce tableau que l'EPS-AKA transmet le trafic minimal par rapport aux autres protocoles. C'est logique puisqu'il présente le plus de vulnérabilités. Les autres protocoles prennent des mesures pour se protéger contre ces faiblesses et cela leur coûte du trafic supplémentaire. Les protocoles EC-AKA et FP-AKA transmet presque le même nombre de bits sur les liaisons radio et backhaul, montante et descendante. Le SE-AKA envoie un trafic nettement supérieur aux autres sur les voies descendantes de la liaison radio et backhaul, puisqu'il est le seul à utiliser le chiffrement asymétrique sur ces voies. Le FP-AKA ne transmet pas les mêmes messages durant le premier accès et les *i* accès ultérieurs effectués. Le nombre de sessions *i* est indiqué par l'opérateur en fonction de la quantité des données à protéger par MHK (qui dépend elle aussi de la crypto période de cette clé). Si l'opérateur par exemple change la clé MHK à chaque 256 Kbits de données (niveau de sécurité très élevé) alors cette clé sera valable pour *i*=47 sessions et la valeur de trafic moyen au cœur sera 1535+(n*640) comme indiqué dans le tableau 4.7.

Pour *n*=6 et *i*=47, les résultats obtenus sont montrés dans la figure 4.20. Il est bien visible que le SE-AKA présente un grand trafic sur l'interface transport dans le réseau cœur. Ceci vient du fait qu'il transmet sur cette interface le certificat de l'utilisateur.

Le protocole FP-AKA montre des résultats intéressants en transmettant beaucoup moins de trafic sur l'interface de transport et sur la voie descendante des deux liaisons radio et backhaul que les deux autres concurrents SE-AKA et EC-AKA.

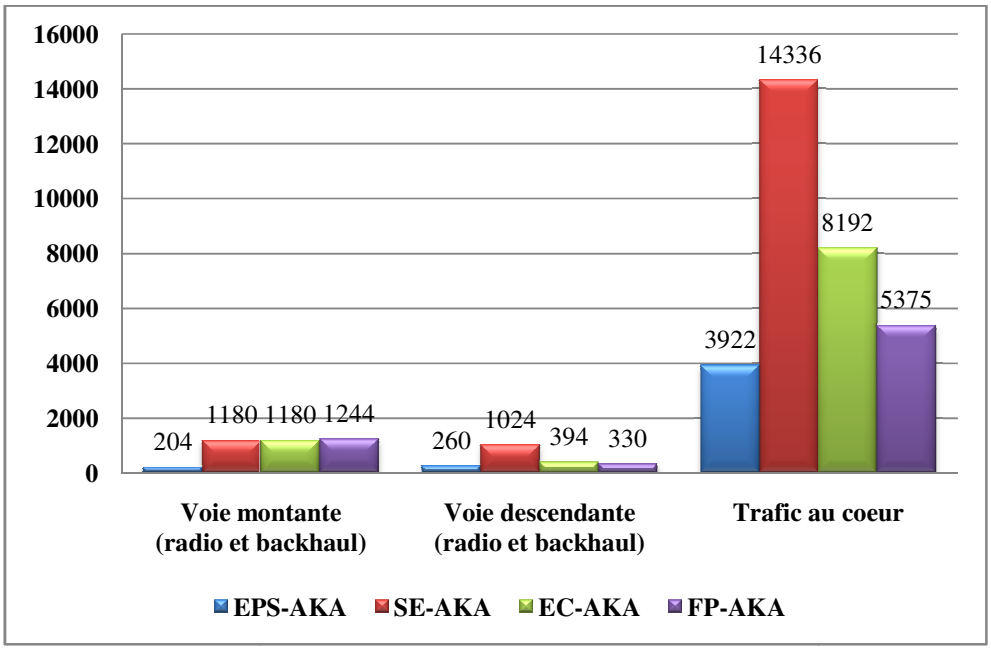

Figure 4.20. Trafic classées selon les interfaces pour les différents protocoles avec n=6 AV

Pour montrer les avantages de notre protocole proposé FP-AKA, calculons maintenant combien il transmet de plus ou de moins de trafic par rapport aux autres protocoles. Le tableau 4.8 montre une comparaison effectuée sur le pourcentage de taux des données de signalisation envoyées par les protocoles étudiés relativement par rapport au FP-AKA. Ce dernier envoie sur la voie radio plus de deux fois que l'EPS-AKA et vingt pour cent de plus sur l'interface cœur, tandis qu'il transmet moins que l'EC-AKA et le SE-AKA. Ce calcul est fait pour n=10.

Pourcentage de trafic supplémentaire	**3GPP EPS-AKA**	**SE-AKA**	**EC-AKA**
Overhead sur l'interface radio (%)	239.22%	-28.58%	0% (égaux)
Overhead sur l'interface coeur (%)	+22.41%	-54.41%	-22.50%
Overhead total (%)	**+36.93%**	**-51.51%**	**-19.51%**

Tableau 4.8. Pourcentage d'overhead de FP-AKA par rapport aux différents protocoles pour n=10 AV

Le nombre de vecteurs AV n est un entier variable et choisit selon l'opérateur. La figure 4.21 montre comment le pourcentage de trafic de signalisation additionnel moyen, envoyé par chaque protocole par rapport au FP-AKA, change en fonction de n. Lorsque n augmente la différence entre le trafic de FP-AKA et celui des autres protocoles diminue. Pour n=10 par exemple, le FP-AKA transmet presque 37 pourcent de trafic de signalisation de plus que le standard EPS-AKA. Pour ce même n, il transmet presque 20 pourcent de moins qu'EC-AKA et plus de 50 pour cent de moins que SE-AKA.

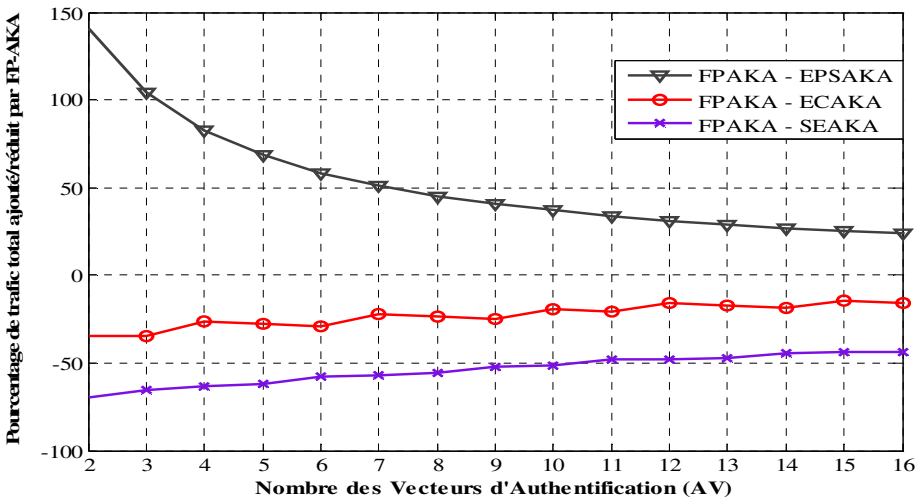

Figure 4.21. Pourcentage de trafic total ajouté/réduit par rapport aux différents protocoles

4.6.4 Délai total de transmission et de traitement

Il est impossible de calculer la durée d'application de chaque protocole AKA puisque les fonctions de sécurité f1,..,f5 ne sont pas connues et puisqu'on ne peut pas estimer la durée de certains traitements effectués. Pour cette raison nous allons comparer maintenant les protocoles en fonction du délai introduit par deux sources : délai de transmission et délai de traitement des tâches essentielles. Chaque part de délai sera étudié séparément. Le délai de transmission dépend de la bande passante effective (congestion du réseau, les conditions radio, la distance entre l'UE et l'eNB, dimensionnement de réseau, etc). Le délai de traitement dépend des ressources de l'UE, de MME et de HSS (la vitesse du CPU, système d'exploitation, la charge sur le système d'exploitation, etc.).

4.6.4.1 Délai de transmission

Les bandes passantes effectives sur les liaisons : montante, descendante et dans le réseau cœur, sont représentées par EBU, EBD et CoreBW définis par :

- EBU (Effective Bandwidth Upload): est la bande passante minimale effective sur la liaison radio et backhaul de la voie montante pour un utilisateur donné;
- EBD (Effective Bandwidth Downlink): est la bande passante minimale effective sur la liaison radio et backhaul de la voie descendant pour un utilisateur donné;
- CoreBW: est la bande passante entre le HSS du réseau d'origine et le MME du réseau servant l'utilisateur.

Pour étudier correctement le délai de transmission, il faut prendre en compte la condition du réseau s'il est condensé ou non, et la situation d'accès d'un utilisateur, s'il est local ou itinérant. Pour cela nous avons classé dans le tableau 4.9 les bandes passantes disponibles dans les différentes situations et conditions [Bou Abdo *et al.*, 2012-b].

Situation d'accès d'un utilisateur et condition du réseau	Bande passante	
Utilisateur local (Bande passante élevé dans le cœur)	CoreBW [20Mb/s , 200Mb/s]	
Utilisateur itinérant (Bande passante faible dans le cœur)	CoreBW [100Kb/s, 10Mb/s]	
Cellule non condensée	EBU [10Mb/s, 50Mb/s]	EBD [20Mb/s, 100Mb/s]
Cellule semi-condensée	EBU [1Mb/s, 9Mb/s]	EBD [2Mb/s, 18Mb/s]
Cellule condensée	EBU [100Kb/s, 900Kb/s]	EBD [200Kb/s, 1.8Mb/s]

Tableau 4.9. Les bandes passantes effectives et disponibles pour un utilisateur

Le délai total de transmission des quatre protocoles est calculé comme suivant :

Délai de transmission total estimé de FP-AKA= [1244/EBU+ 330/EBD+ (1535+ (n*640)/Core BW].

Délai de transmission total estimé d'EPS-AKA= [204/EBU+ 260/EBD+ (82+(n*640))/Core BW].

Délai de transmission total estimé de SE-AKA= [1180/EBU+ 1024/EBD+ (2048+ ceil((n*640+ 8252) /1024)*1024)/CoreBW].

Délai de transmission total estimé d'EC-AKA= [1180/EBU+ 394/EBD+ (3072+ ceil((n*640+ 399)/1024)*1024)/CoreBW].

Pour faciliter la comparaison, nous avons calculé le taux relatif de délai supplémentaire de notre protocole FP-AKA avec chacun des autres protocoles. Ce calcul a été effectué pour n=10 et dans des différentes conditions et situations. Les résultats obtenus sont donnés dans le tableau 4.10.

Différence dans le délai de transmission de FP-AKA vs (n=10)	EPS-AKA		EC-AKA		SE-AKA	
	Utilisateur Local	**Utilisateur itinérant**	**Utilisateur Local**	**Utilisateur itinérant**	**Utilisateur Local**	**Utilisateur itinérant**
Cellule non condensée	+28.76us (50.39%)	+178us (25.47%)	-14.1%	-21.85%	-43.86%	-53.69%
Cellule semi-condensée	+126.7us (182%)	+276us (38.63%)	-13.824%	-19.28%	-28.65%	-50.74%
Cellule condensée	**+1.2ms (297%)**	**+1.35ms (131.43%)**	+1.51%	-7.63%	-18.39%	-34.84%

Tableau 4.10. Différence de délai de transmission entre FP-AKA et les autres protocoles

Du tableau 4.10, on peut déduire que le FP-AKA possède les délais de transmission minimale par rapport aux deux autres protocoles EC-AKA et SE-AKA. Son délai de transmission présente par exemple jusqu'à 21.85% de moins par rapport à celui d'EC-KA et 53.69 % par rapport à celui de SE-AKA dans les cellules non condensées et pour un utilisateur itinérant. Il est normal que notre protocole présente des délais supérieurs à ceux du standard EPS-AKA puisqu'il envoie plus de données afin d'assurer une robustesse parfaite contre les attaques étudiées. Notre protocole présente par exemple 182% plus (par rapport à EPS-AKA) sur le délai de transmission, dans le cas d'une cellule semi-condensée, mais son délai (126.7 us) reste très petit, très acceptable et ne cause pas des problèmes pour les services de l'abonné.

4.6.4.2 Délai de traitement au niveau des entités

Puisque les ressources de calcul du mobile sont beaucoup plus limitées que celles des nœuds du réseau, nous allons considérer le délai du traitement du côté mobile et seulement le délai du traitement de la tâche principale et la plus lourde au niveau du MME et HSS et qui est le chiffrement/déchiffrement.

Pour calculer le délai de traitement (chiffrement, déchiffrement, vérification de l'intégrité des messages, etc.) chez l'UE, nous avons utilisé les données du tableau 4.11 offert par la librairie Crypto++ [Benchmarks, 2013]. Ce tableau présente les vitesses de calcul de quelques algorithmes cryptographiques utilisés, en supposant que nous disposons d'un processeur Intel Core 2, 1.83 GHz, sous Windows Vista en mode 32-bits.

Algorithme	Traitement
EEA=AES/CTR (128-bits)	139 MB/s, avec 0.689us pour vérifier la taille de la clé
EIA=AES/CMAC (128 bits)	109 MB/s, avec 0.600us pour vérifier la clé
SHA-256	111 MB/s
RSA 1024 (chiffrement)	0.08 ms/bloc

Tableau 4.11. Vitesse de calcul des algorithmes utilisés

Connaissons déjà la taille des données à chiffrer/déchiffrer et à vérifier, le délai de traitement calculé au niveau de l'UE et au niveau de MME et HSS pour les trois protocoles non-standard, est donné dans le tableau 4.12. Notons que le protocole standard, EPS-AKA, ne chiffre aucun message de signalisation et c'est la raison pour laquelle il souffre de multiples vulnérabilités.

	Délai de traitement	
	UE	MME / HSS (chiffrement de *10* AV)
FP-AKA	84.37us (**-94.57%** par rapport à SE-AKA)	6.4us (**-98.85%** par rapport à EC-AKA et SE-AKA)
EC-AKA	83.5us	0.56ms
SE-AKA	1.54ms	0.56ms

Tableau 4.12. Délai de traitement des messages au niveau de l'UE, MME et HSS

Comme nous remarquons d'après le tableau 4.12, les deux protocoles FP-AKA et EC-AKA ont presque le même délai de traitement au niveau de l'équipement mobile avec 94.57% moins de celui de SE-AKA. Ce délai diminuera avec les nouveaux équipements mobiles et leurs processeurs si puissants.

Côté réseau, la tâche qui exige le plus grand traitement est le chiffrement des n vecteurs AV. Dans notre protocole FP-AKA, le chiffrement/déchiffrement symétrique entre MME et HSS des vecteurs AV (par l'algorithme AES/CTR) réduit le délai de traitement au niveau de ces entités à 98.85% par rapport à EC-AKA et SE-AKA qui utilisent tous les deux le chiffrement asymétrique. Par exemple et comme indiqué dans le tableau 4.12, pour n=10 AV, le chiffrement asymétrique des 7 blocs RSA par EC-AKA et SE-AKA prend 0.56ms alors qu'avec FP-AKA ca prend 6.4us.

Notons que HSS et MME disposent des processeurs qui sont une dizaine de fois plus rapide que le CPU core 2 utilisé pour obtenir les résultats montrés dans le tableau 4.12. Dans ce cas les délais de traitement montrés dans le tableau au niveau de ces deux entités seront énormément réduits, mais notre protocole garde toujours le même pourcentage de réduction de délai.

Comme délai total de transmission et de traitement, notre protocole présente donc le délai total minimal par rapport aux protocoles EC-AKA et SE-AKA. Dans le pire des cas (c.à.d. dans une cellule condensée), FP-AKA ajoute au standard EPS-AKA 1.28 ms de délai total pour un utilisateur local et 1.43 ms pour un utilisateur en itinérance. Ce délai est très petit par rapport au délai total de la procédure AKA et d'établissement de la sécurité NAS et AS (délai de transmission des messages de signalisation, délai de génération des vecteurs, délai de dérivation des clés, etc.). Un UE qui allume son mobile et qui reçoit son GUTI et ses clés EPS ne va pas sentir du tout de ce délai supplémentaire qui est presque négligeable.

Note : l'ECC ajoute plus de 70 fois de délai de traitement sur le mobile (ainsi que le HSS et le MME) puisque selon [Benchmarks, 2013] on met 5.65 ms pour traiter un seul bloc chiffré par la courbe elliptique au lieu de 0.08 ms par RSA. Le délai de traitement au niveau d'UE devient 5.653 ms (consommation de batterie beaucoup plus grande que RSA) au lieu de 84.37us.

4.6.6 Résumé des résultats

Nous présentons dans le tableau 4.13, le résumé comparatif des 4 protocoles étudiés. Ce tableau donne un classement des protocoles pour chaque paramètre (sécurité, coût, taux de données

ajoutées sur la signalisation, et délai total) où on donne la valeur ‘4’ pour signifier ‘le plus faible’, et la valeur ‘1’ signifie ‘ le meilleur’.

Paramètres de QoS	FP-AKA	EC-AKA	SE-AKA	Standard EPS-AKA
Sécurité	1	2	3	4
Coût	1	1	3	1
Taux de données ajoutées (Overhead)	2	3	4	1
Délai total	2	3	4	1

Tableau 4.13. Fiche technique évaluant la QoS de FP-AKA, EC-AKA, SE-AKA, et EPS-AKA

Il est clair que notre protocole FP-AKA a le meilleur niveau de sécurité et de coût. Le standard AKA de 3GPP est le plus rapide puisqu’il n’implémente aucune sécurité supplémentaire à son niveau, sans oublier que son niveau de sécurité est le plus faible (mauvais). Dans notre conception, la sécurité est un facteur très important et elle nécessite une augmentation acceptable des ressources. L’EC-AKA a une performance acceptable par rapport à SE-AKA qui a le plus mauvais performance sur l’ensemble des paramètres, il est donc considéré comme non adéquat pour les futures implémentations.

Puisque FP-AKA est le seul protocole satisfaisant les exigences de la sécurité du réseau EPS en atteignant d’excellentes performances de QoS, il sera un sérieux candidat pour remplacer le mécanisme d’authentification et d’accord des clés standard en EPS.

4.7 Conclusion

Dans ce chapitre, nous avons étudié les faiblesses de la sécurité du réseau EPS, dans le but de proposer des solutions pour toutes ces faiblesses avec un coût minimal et une meilleure QoS.

Le premier point faible abordé dans ce chapitre a été la transmission de l’IMSI en claire. Nous avons vu que l’interception de cette identité par un ‘IMSI catcher’ ou par d’autres moyens permet l’identification et le traçage des utilisateurs. Les solutions existantes à traiter ce problème ont été proposées par Al-Saraireh et Caragata (Enhanced EMSUCU) qui ont offert une protection par chiffrement symétrique de l’IMSI. Nous avons analysé la solution de Caragata (variante améliorée d’Al-Saraireh) et nous avons proposé deux améliorations essentielles à cette solution qui la rend plus robuste. La première consiste à limiter la recherche dans le HSS aux abonnés d’un MME spécifique ou aux mobiles éteints, et la deuxième consiste à contrecarrer les attaques possibles sur l’EEMSUCU et surtout le DoS. Nous avons vu que le coût d’identification d’un abonné (recherche et nombre d’opérations) après notre proposition est négligeable par comparaison avec l’EEMSUCU.

Ensuite nous avons analysé le protocole standard de 3GPP EPS-AKA, et nous avons identifié ses vulnérabilités, ainsi que les scénarios qui permettent les attaques à casser ce protocole. Nous avons montré que les risques résultant par l’exploitation de ces faiblesses peuvent : empêcher les utilisateurs d’accéder au réseau, compromettre la clé permanente K, bloquer les services de

l'opérateur mobile, provoquer l'usurpation des identités des utilisateurs ou de MME. Pour chacune des attaques identifiées, nous avons proposé des solutions convenables.

Les meilleurs protocoles AKA proposés pour éviter ces attaques et remplacer le protocole EPS-AKA ont été crypte-analysé, chacune a été exploitée par une ou plusieurs vulnérabilités à travers une attaque spécifique. Le deuxième AKA était le protocole SE-AKA qui a été crypte-analysé en utilisant l'attaque par dictionnaire, et les attaques par rejoue, déni de service et MITM. Le troisième AKA était le protocole EC-AKA qui a été crypte-analysé par l'attaque par rejoue, déni de service sur le HSS/Auc et l'attaque sur les réponses des données d'authentification.

Nous avons proposé un nouveau protocole capable de résoudre toutes les faiblesses relevées dans les trois protocoles étudiés. Ce protocole a été appelé 'Full Protection Authentication and Key Agreement' FP-AKA. Il assure une protection complète de la procédure AKA et de la sécurité EPS contre toutes les attaques possibles. Ce protocole a été présenté avec toutes ses composantes de sécurité.

Une étude comparative entre FP-AKA, EC-AKA, SE-AKA et l'EPS-AKA selon quatre paramètres de QoS a été réalisée. Les paramètres étudiés sont: la sécurité, le coût (CAPEX-OPEX), le taux de données ajoutées sur la signalisation, et le délai total. Le FP-AKA a obtenu les meilleurs résultats dans les deux premiers paramètres et atteint de très bons résultats dans les paramètres restants.

Puisque FP-AKA est le seul protocole satisfaisant les exigences de sécurité et les spécifications de l'EPS, et grâce à son excellente performance selon les différents paramètres QoS, il peut être considéré comme étant un excellent protocole AKA.

Conclusion générale

Conclusion générale

Ces travaux de thèse ont permis, d'abord, de montrer les failles de sécurité dans les communications IP multicast par DVB satellitaire, et dans les réseaux de téléphonie mobiles 3G et 4G. Ensuite, différentes solutions, à ces vulnérabilités, ont été proposées. À savoir un système de sécurité basé sur les séquences chaotiques, et un nouveau protocole de sécurité, qui apportent une nette amélioration dans la robustesse des systèmes de communication étudiés. Les différents services de la sécurité concernés par les solutions proposées sont: l'authentification, la confidentialité, l'intégrité et la disponibilité.

Dans le premier chapitre, nous avons introduit les concepts de base nécessaires pour la compréhension des sujets traités dans la thèse, comme la sécurité de l'information, les fonctions de sécurité nécessaires, les attaques possibles, le chiffrement symétrique et à clé publique, la gestion des clés secrètes, les signaux chaotiques et les protocoles.

Dans le deuxième chapitre, nous avons étudié la sécurité et le transfert des communications IP multicast à travers le satellite DVB ainsi que la question de l'évolutivité du groupe multicast. D'abord, nous avons présenté les méthodes d'encapsulation utilisées dans le système DVB et nous avons sélectionné l'ULE qui est la meilleure solution existante. Ensuite, nous avons présenté les requis demandés par l'ULE pour opérer avec les approches de commutation (label-switching et self-switching) afin d'assurer le transfert efficace des paquets multicast. Après, nous avons analysé les solutions de sécurité utilisées dans les systèmes actuels des communications IP par DVB-S et nous avons pointé ses faiblesses. Puis, nous avons proposé un système de sécurité très performant. Ce système s'appuie sur : une méthode d'encapsulation améliorée d'ULE (EULE) qui offre les informations nécessaires : pour la commutation ; un mécanisme de sécurité performant qui authentifie chaque trame EULE et la chiffre avec son code MAC à l'aide d'une clé différente ; un système de gestion de clés à deux couches LKH (TLKH) qui traite le problème d'évolutivité, des fonctions chaotiques pour la génération des clés et le chiffrement, un paquet spécial KPDU pour le transport des clés, et un message d'alarme DULM pour rétablir la synchronisation entre le fournisseur et le RCST.

Ensuite, nous avons analysé les performances de ce système et nous avons montré sa supériorité sur tous les niveaux, moyennant une faible modification de la trame ULE. En première étape, nous avons montré les avantages de TLKH par rapport aux autres systèmes de gestion des clés existants, surtout en termes de réduction du coût de rekeying sur la liaison satellitaire. En deuxième étape, nous avons montré que ce système assure une grande réduction en consommation de la bande passante pour les données de gestion des clés par rapport à la meilleure méthode utilisée ECPVSS, et un coût presque négligeable pour le taux de données ajoutées en utilisant l'approche label-switching.

Dans le troisième chapitre, nous avons présenté les architectures des réseaux de communications mobile UMTS et EPS et nous avons étudié en détails la sécurité dans ces réseaux. Dans cette étude, nous avons montré les éléments robustes de la sécurité 3G qui sont conservés dans la sécurité EPS, et les améliorations (ou les nouveautés) apportées sur l'architecture de la sécurité EPS. Nous avons analysé les exigences répandues de la sécurité EPS et celles qui restent sans traitement complète. Il s'avère aussi de cette étude que l'aspect le plus sensible de la sécurité EPS est l'accès sécurisé au réseau. Nous avons présenté brièvement ses quatre composantes

essentielles: l'identification des utilisateurs et des terminaux, l'authentification et l'établissement des clés, via la procédure EPS-AKA, la dérivation des nouvelles clés, la protection de l'intégrité de la signalisation NAS, AS et le chiffrement de la signalisation et des données. Ces composantes constituent le contexte de toute analyse de la sécurité de l'EPS.

Dans le quatrième chapitre, nous avons identifié les faiblesses de la sécurité EPS, et nous avons proposé un nouvel protocole qui a permis de résoudre ces faiblesses avec un coût minimal et une meilleure QoS. D'abord, nous avons abordé la question de la transmission en clair de l'IMSI, et à ce sujet nous avons présenté et analysé les solutions proposées par Al-Saraireh et Caragata (Enhanced EMSUCU). Nous avons montré que la solution de Caragata n'est pas efficace et qu'elle est vulnérable aux attaques. Afin de la rendre robuste, nous avons proposé deux améliorations essentielles : le coût d'identification d'un abonné (recherche et nombre d'opérations) est devenu négligeable ; la résistante aux attaques est efficace.

Ensuite, nous avons analysé le protocole EPS-AKA, et avons identifié ses vulnérabilités ainsi que les attaques contre ce protocole (attaque de déni de service contre l'UE et le HSS/AuC, attaques sur les réponses des données d'authentification, etc.). Nous avons montré que le risque résultant de ces attaques peut : empêcher les utilisateurs d'accéder au réseau, provoquer l'usurpation des identités des utilisateurs ou de MME, etc.

Ensuite nous avons analysé les meilleurs protocoles AKA proposés dans la littérature pour éviter les attaques citées au paragraphe précédent. Cependant, des vulnérabilités existent toujours pour chacune des solutions à travers une attaque spécifique (attaque par dictionnaire, attaque par rejoue, etc.). Ensuite nous avons proposé notre protocole FP-AKA qui a permis de renforcer toutes les faiblesses relevées, et nous avons montré comment il a assuré la protection de la procédure AKA et de la sécurité EPS contre tous les attaques. Enfin, l'étude comparative que nous avons effectué entre le protocole FP-AKA et les autres protocoles AKA (EC-AKA, SE-AKA, EPS-AKA) sur quatre paramètres de QoS a montré l'intérêt de notre protocole. L'excellente performance du FP-AKA le place comme étant le meilleur protocole AKA proposé à ce jour.

Annexes

Annexes

Annexe A : IPsec (Internet Protocol Security)

Le protocole IPSec est un protocole au niveau réseau qui ajoute des services de sécurité au protocole IP (voir figure A.1).

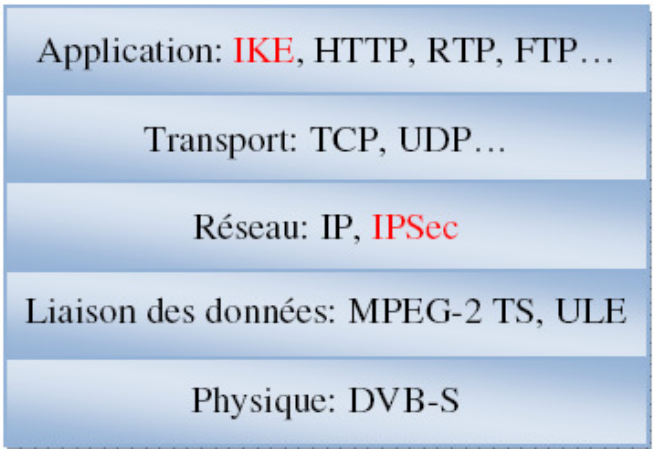

Figure A.1. Modèle TCP/IP pour DVB-S sécurisé avec IPSec

Les services de sécurité offerts sont: la confidentialité, l'intégrité des données et l'authenticité de la source. Chaque paquet IP peut être soit authentifié, soit chiffré, soit les deux à la fois.

IPSec est, en effet, composé d'une série de protocoles. Les plus importants sont :

- *AH (Authentication Header)* : fournit l'authentification d'origine, et l'intégrité des données.
- *ESP (Encapsulating Security Payload)* : fournit la confidentialité, l'authentification d'origine et l'intégrité des données [RFC 4303, 2005].
- *IKE (Internet Key Exchange)* : permet l'établissement d'une Association de Sécurité (SA – Security Association) par la négociation des clés, des protocoles et des algorithmes qui seront utilisés. IKE se base sur ISAKMP (Internet Security Association Key Management Protocol) et les algorithmes Oakley et SKEME [RFC 4306, 2005].

A.1 Association de sécurité SA

Une SA [RFC 2401, 1998] est un concept de base pour IPSec. Une SA englobe l'ensemble d'algorithmes et des paramètres (par exemple : clés, périodes de validités des clés) utilisés pour les services de sécurité. Une SA est unidirectionnelle, alors, pour sécuriser un trafic bidirectionnel, deux SA sont nécessaires, une pour chaque direction. La SA d'un certain paquet IP est définie par le SPI (Security Parameter Index), un champ de l'en-tête IPSec qui est un index pour la SAD (Security Association Database), et l'adresse destination. Plusieurs SA peuvent être utilisées, chacune avec son propre SPI, ce qui permet d'avoir plusieurs niveaux et ensemble de sécurité pour une certaine liaison.

A.2 Modes : tunnel, transport

IPSec peut être utilisé en 2 modes différents : mode tunnel et mode transport. En Mode tunnel, tous les paquets IP originaux sont chiffrés/authentifiés et un nouveau en-tête IP est créé pour chaque paquet. Ce mode est utilisé pour les VPN (Virtual Private Networks), il cache les caractéristiques du trafic (voir figure A.2).

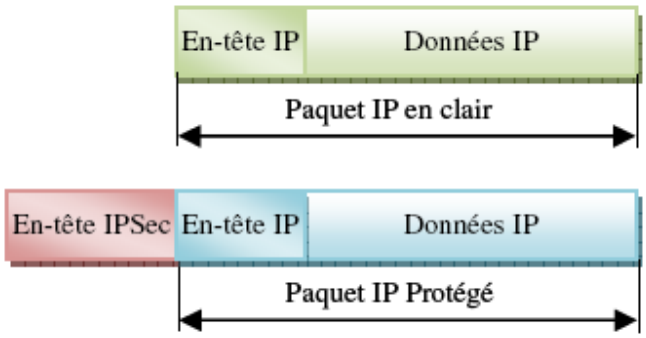

Figure A.2. IPSec en mode tunnel

En mode transport, l'en-tête original IP est utilisé tel qu'il est et le chiffrement concerne seulement les données du paquet IP (voir figure ci-dessous).

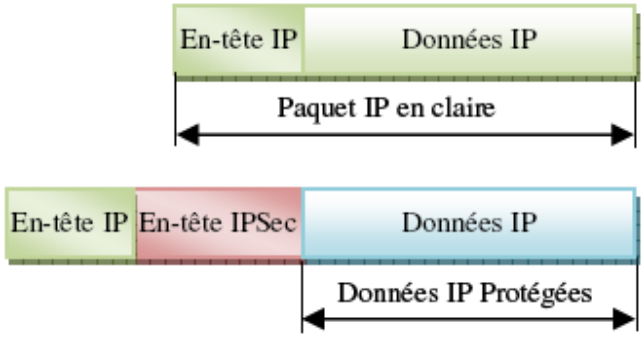

Figure A.3. IPSec en mode transport

A.3 Bases des données

IPSec utilise deux bases de données :

- SPD (Security Policy Database): contient les données qui permettent d'indiquer les services de sécurité requis pour chaque paquet IP traité par IPSec. Sa structure est décrite en [RFC 4301, 2005].

- SAD (Security Association Database): contient tous les SA établies. SPD est la première base de données à être consultée. Si elle contient une SA qui satisfait les requis de la SPD, cette SA sera alors utilisée. Sinon, IKE est utilisé pour créer une nouvelle SA.

A.4 IKE

La gestion des clés est réalisée avec un protocole spécifique, l'IKE. Il est utilisé pour créer les SA et les actualiser. Son fonctionnement est décrit dans la figure ci-dessous :

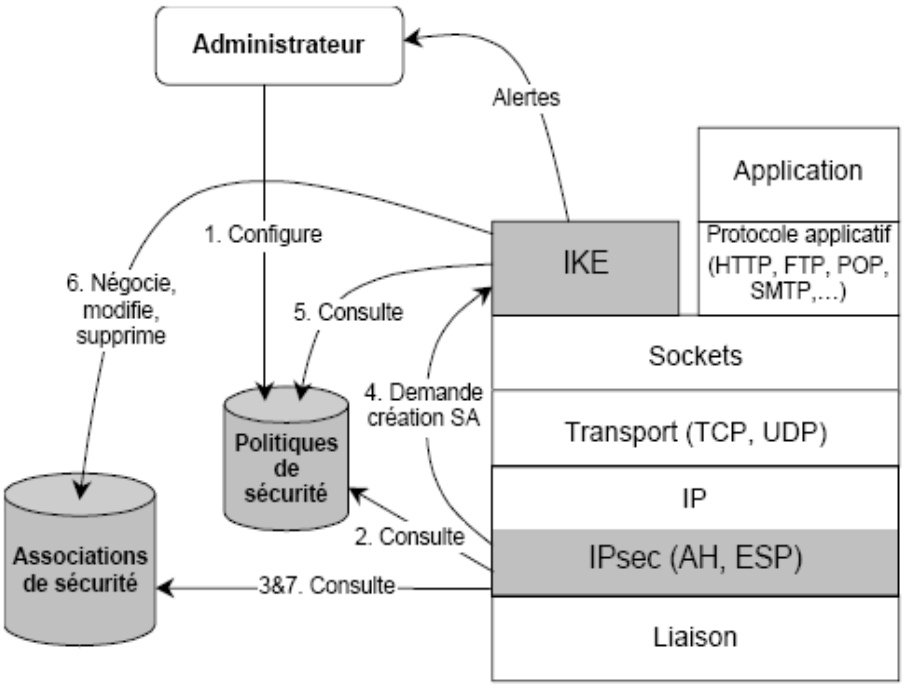

Figure A.4. Protocole IKE

La première observation qu'on peut faire est que l'IKE est un protocole de niveau application, tandis que l'IPSec se situe au niveau réseau.

Pour que le système puisse fonctionner, l'administrateur doit, dans une première étape, configurer la SPD (1). Ensuite, pour chaque paquet IP qui doit être envoyé, IPSec consulte la SPD pour voir les services de sécurité requis pour cette connexion (2). Après, il vérifie s'il y a une SA dans le SAD qui supporte les requis (3). S'il n'y a pas, il envoie une demande à l'IKE (4). Celui-ci consulte le SPD (5), établie une nouvelle SA et l'ajoute au SAD (6). En fin, IPSec consulte à nouveau le SAD pour retirer la SA (7) et l'utilise pour appliquer les protocoles et algorithmes adéquats.

Les échanges des messages IKE sont indépendants d'IPSec et se déroulent en deux phases. La première phase permet l'établissement d'une SA propre à IKE. Contrairement aux SA d'IPSec, qui sont unidirectionnelles, cette SA est bidirectionnelle. Elle permet les échanges de la deuxième phase qui établissent les SA d'IPSec. Les messages de cette deuxième phase sont protégés par les protocoles, les algorithmes, et les clés de SA de l'IKE qui a été établie dans la première phase.

La première phase peut être exécutée en deux modes: *mode principal (main mode)* et *mode agressif (aggressive mode)*. Le mode principal, consiste en 6 messages. Le mode agressif compacte l'information qui sera échangée en 3 messages seulement.

IKE utilise le protocole ISAKMP. Celui-ci a été conçu pour être souple et pour être utilisé par d'autres applications. Ses messages sont constitués d'un chaînage des blocs. Il existe 13 types de blocs possibles:

- *SA* (*Security Association*) : indique le contexte de l'échange en première ou deuxième phase, ainsi le fait qu'il s'agit d'un échange IPSec.
- *P* (*Proposal*) : contient des propositions pour le SA : mécanisme à utiliser (AH ou ESP), et SPI à associer au SA.
- *T* (*Transform*) : est un bloc complément au bloc *Proposal* et contient des propositions sur les algorithmes de chiffrement ou fonctions de hachage à utiliser pour la SA.
- *KE* (*Key Exchange*) : transporte les données nécessaires à la génération de la clé de session.
- *ID* (*Identification*) : transporte les données utilisées pour l'identification des tiers.
- *CERT* (*Certificate*) : utilisé pour le transport des certificats (s'ils sont utilisés)
- *CR* (*Certificate Request*) : demande un certificat.
- *HASH* (*Hash*) : résultat d'une fonction de hachage.
- *SIG* (*Signature*) : résultat d'une fonction de hachage signée.
- *NONCE* (*Nonces*) : transporte des aléas (des données utilisées qu'une seule fois)
- *N* (*Notification*) : permet l'échange des messages d'erreur ou d'information sûr l'état actuel des négociations.
- *D* (*Delete*) : spécifie qu'une SA sera effacée.
- *VID* (*Vendor ID*) : permet à deux installations de même marque de se reconnaître pour pouvoir utiliser des implémentations propres.

Annexe B1 : Les échecs d'authentification

Les échecs d'authentification [TS 24.301, 2011] qui peuvent arrivés durant la procédure AKA suite à la différence entre les valeurs reçus et les valeurs attendus sont listés ci-dessous :

- *Échec du code MAC* : Si l'USIM détermine que le code MAC reçu est différente de XMAC, il indique ca au ME, qui envoie un message d'échec d'authentification (*Authentication Failure message*) vers le MME avec une indication de la cause.

- *Échec de synchronization:* Cela se produit lorsque l'USIM détermine que le numéro de séquence reçu n'est pas dans la gamme correcte. Le comportement de l'USIM et de l'AUC dans ce cas est le même pour l'UMTS AKA et l'EPS AKA. L'USIM calcule un paramètre AUTS comme indiqué dans la figure B1.1 et l'inclut dans un message d'échec de la synchronisation (*Synchronisation failure message*) pour l'envoyer au MME. Le MME transmet l'AUTS au HSS en demandant de nouveaux vecteurs d'authentification. L'AMF utilisée pour calculer le code MAC-S est fixé à zéro donc il n'a pas besoin d'être transmis au HSS. Le HSS/AuC utilise l'AUTS pour synchroniser le SQN_{HE} stocké dans le serveur HSS/AuC avec le SQN_{MS} inclut dans l'AUTS. Le HSS doit informer de nouveau l'AuC que la demande est liée à l'EPS pour qu'il remette à '1' le bit d'AMF. Après la synchronisation du SQN_{HE} (c.à.d. la remise du SQN_{HE}= SQN_{MS}), le HSS génère de nouveaux vecteurs EPS et les envoie au MME.

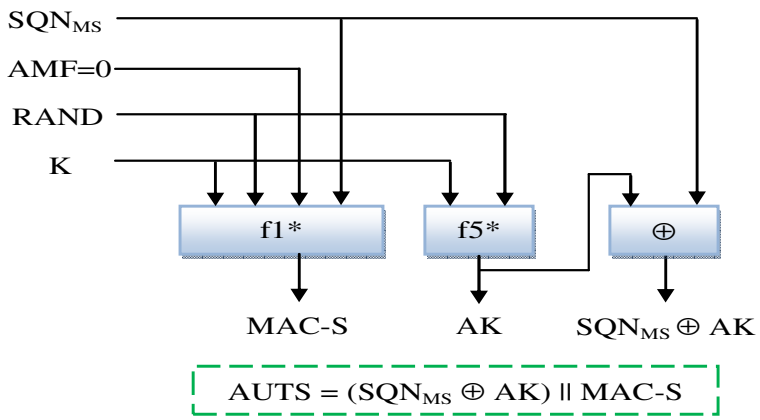

Figure B1.1. Construction du paramètre AUTS pour la resynchronisation

- *Type incorrect du vecteur d'authentification* : Si la vérification du 'bit de séparation AMF' dans le ME échoue, le ME envoie un message d'échec d'authentification (*Authentication Failure message*) au MME avec une indication de la cause.

- *Réponse d'authentification invalide* (Invalid authentication response) : Si le MME détermine que XRES est différente de RES alors, selon le type de l'identité utilisée, le MME peut décider d'initier une nouvelle identification et procédure d'authentification vers l'UE, ou bien il peut envoyer un message de rejet d'authentification (*Authentication Reject message*) à l'UE et abandonne la procédure.

Notant que dans EPS-AKA, il n'est plus nécessaire d'envoyer des *rapports d'échec d'authentification* du MME au réseau d'origine comme dans UMTS-AKA où le VLR/SGSN envoie ces rapports au HLR [TS 33.102, 2012] dans tous les cas d'échecs d'authentification. La raison de ce changement c'est que l'utilité de ces rapports s'est avérée assez limité.

Annexe B2 : Différences entre la sécurité UMTS et EPS

Nous avons étudié dans le chapitre 3 le fonctionnement de la sécurité dans les réseaux de téléphonie mobile de troisième génération UMTS et dans les réseaux de quatrième génération EPS. D'après cette étude approfondie de la sécurité, nous pouvons conclure les différences ou les nouveautés apportées par les fonctions de sécurité EPS par rapport à l'UMTS comme indiqué dans le tableau B2.1

Fonctions	**Comparaison de la sécurité entre UMTS et EPS**	
	3G/UMTS	4G/EPS
Identité de l'utilisateur	Identité permanente et temporaire : IMSI et P-(TMSI)	Identité permanente et temporaire : IMSI et GUTI
Identité du terminal	IMEI, IMEISV transmis en clair	IMEI, IMEISV est chiffré
Authentification mutuelle entre l'UE et le réseau et établissement d'une nouvelle clé (AKA)	Pas d'authentification du réseau de service, seulement assurance que le réseau de service est autorisé par le réseau d'origine.	Authentification du réseau de service
	Etablissement des clés CK, IK entre l'UE et le VLR/SGSN. CK, IK sont envoyés ensuite au RNC qui doit effectuer le chiffrement et l'intégrité.	Etablissement d'une clé maitre K_{ASME} qui sera utilisé pour dériver d'autres clés entre l'UE et le MME. K_{ASME} ne quitte jamais MME.
	Vecteur d'authentification UMTS: (RAND, XRES, CK, IK, AUTN)	Vecteur d'authentification EPS: (RAND, XRES, K_{ASME}, AUTN)
	L'AuC génère l'UMTS AV	Le HSS génère l'EPS AV à partir de l'UMTS AV
		Une fonction de dérivation de clés est implémentée dans le ME et dans le HSS pour dériver la clé K_{ASME}.
	Le bit le plus signifiant du champ AMF='0' pour les utilisations existantes	Le bit le plus signifiant du champ AMF='1' pour indiquer que le vecteur d'authentification est pour 'usage EPS'
Niveau (s) de protection	Un seul niveau de protection AS	2 niveaux de protection AS et NAS
Confidentialité et intégrité des données	Confidentialité des données usager et de signalisation, et intégrité des données de signalisation se fait dans l'UE et le RNC	Confidentialité des données usager et de signalisation AS et intégrité des donnes AS se fait dans l'UE et l'eNB
	Pas de protection NAS	Chiffrement et intégrité des données NAS entre UE et MME

Point de terminaison de RRC et usager	RNC est mis dans une place sécurisée	Introduction de la sécurité de l'eNB, avec la définition de toutes ces exigences (puisqu'elle est positionnés dans des places exposés et en dehors du domaine de sécurité de l'opérateur)
Sécurité dans le domaine réseau	IPSec ESP en mode tunnel avec l'IKE sont utilises pour la sécurité des messages entre le RNC et le réseau de service (VLR/SGSN). Le MAPSec est utilisé pour la sécurité de signalisation entre le VLR/SGSN et les bases de données (HLR, EIR)	IPSec ESP en mode tunnel (mode transport est facultatif) avec IKEv2 sont utilisés pour la sécurité des messages échangés entre les éléments du réseau EPS (entre l'eNB et le MME/S-GW)
Les clés cryptographiques	Deux clés seulement : CK, IK pour le chiffrement et l'intégrité d'AS	-Nouvelle hiérarchie des clés. -Cinq clés pour le chiffrement et l'intégrité proviennent de K_{ASME}, a) K_{NASenc}, K_{NASint} pour la protection de la signalisation NAS. b) K_{eNB} pour dériver les clés AS : K_{RRCenc}, K_{RRCint} (pour la protection de la signalisation RRC) et K_{UPenc} (pour chiffrer les données usagers).
Algorithmes de sécurité	UIA et UEA : KASUMI (algorithme cassé) et SNOW 3G.	EIA, EEA : SNOW 3G, AES et ZUC.
Dérivation de clés	Pas de KDF autre que f3, f4 et f5	f3, f4, f5 + KDF basé sur HMAC-SHA-256 pour la dérivation de toutes les clés EPS

Tableau B2.1 Différences principales entre la sécurité de l'UMTS et de l'EPS

Annexe B3: L'algorithme à clé publique RSA-OAEP

Dans cette variante de RSA, l'émetteur qui voudrait envoyer le message X, ajoute à ce message une entrée supplémentaire *AI* (Additional Information) afin de transmettre les deux (X avec *AI*) chiffrés par RSA-OAEP. Dans notre protocole FP-AKA, X=A, B ou C1.

Chaque entité doit savoir la forme et le contenu de *AI* avant d'être requis par le système. Le but de *AI* consisterait à permettre à l'émetteur d'indiquer qu'il a l'intention d'employer le message dans un contexte défini et d'attacher le message à ce contexte. La méthode pour la mise en forme et la distribution de *AI* est définie par l'application. *AI* peuvent comprendre une représentation des informations partagées, soit échangés par les deux entités, soit obtenus à partir des protocoles du niveau supérieur. *AI* peuvent être:

- Les noms ou d'autres informations d'identification de l'émetteur et du récepteur ;
- Nonce ou une valeur de compteur ;
- Le type, la longueur, ou l'utilisation prévue du message ;
- données secrètes ou autres données publiques, partagées par les entités ;

- Une chaine vide (empty string).

Après l'accord de la forme et du contenu de *AI* et après la réception de la clé publique de chacune des entités. L'émetteur génère le message X, de taille ≤ *nlen–2.hlen–2* où *nlen* et *hlen* sont les tailles du modulo *n* et de la sortie de la fonction de hachage en bytes. L'émetteur ajoute à ce message l'entrée *AI.* Ensuite, il utilise RSA-OAEP pour chiffrer (*X* et *H(AI)*). Le RSA-OAEP ajoute du bourrage sur (*X* et Hash(*AI*)) puis il les masque avant de les chiffrer par la clé publique RSA du récepteur. Pour faire ceci, RSA-OAEP utilise une fonction de hachage pour obtenir le H(*AI*) et une fonction de génération de masque (MGF) afin d'obtenir le masque pour masquer le message (*X* et H(*AI*)), qui sera ensuite chiffré par RSA afin d'obtenir le message chiffré CT (Cipher Text). Ce dernier est envoyé au récepteur, qui à son tour utilise RSA-OAEP pour déchiffrer CT en utilisant sa clé privée. Dès que le récepteur obtient *X* et H(*AI*), il utilise l'information partagée *AI* et calcule le hachage de *AI* afin de comparer la valeur reçu H(*AI*) à celui calculé. Si le récepteur trouve que les deux valeurs sont égales, alors il obtient un genre d'assurance sur l'intégrité des données reçus. Les détails du chiffrement RSA-OAEP sera donné dans la section suivante. Mais, avant de commencer le processus de chiffrement, l'émetteur et le récepteur doit se mettre en d'accord sur une fonction de hachage approuvé et approprié pour l'appliquer sur *AI*, ainsi qu'une fonction de hachage pour l'utiliser avec la fonction MGF. Ces deux fonctions sont utilisées par RSA-OAEP qui peut choisir SHA-256 comme une fonction de hachage approuvé.

B3.1 Chiffrement par RSA-OAEP

Pour savoir comment se fait le chiffrement par RSA-OAEP sur le message X et *AI* (*AI* peut être une chaine vide, et le hachage s'applique ainsi sur une chaine vide [Jonsson et Kalisiki, 2003]) on a représenté cette paragraphe. D'abord RSA-OAEP utilise les composantes suivantes:

- Une fonction de hachage approuvé (SHA-256) pour l'appliquer sur *AI*.
- Un RBG approuvé (la fonction f0), pour générer le *mgfseed.*
- Une fonction MGF (Mask Generation Function) qui prend comme entrée les deux paramètres (*mgfseed*, *masklen*), où *masklen* est la taille du masque (en bytes) attendu à la sortie de cette fonction. L'entrée *mgfseed* est une entrée aléatoire qui peut avoir une taille plus petite ou plus grande que la taille du masque à générer. Le MGF utilise une fonction de hachage pour générer un masque (série de bits) de longueur *masklen,* et qui est utilisé pour masquer une autre série de bits.
- RSAEP (RSA- Encryption Primitive) et RSADP (RSA- Decryption Primitive) sont utilisés pour chiffrer avec RSA les donnés masqués et déchiffrer par RSA ces donnés.

Une fois l'émetteur (l'UE, le MME ou le HSS) crée le message X et possède *AI* (optionnel), il applique le chiffrement RSA-OAEP pour chiffrer ces deux paramètres et obtenir CT (Cipher Text). Voyons dans la figure B3.1 comment RSA-OAEP fonctionne. Le processus de chiffrement commence en entrant *AI* à une fonction de hachage approuvé, qui donne à sa sortie H(*AI*) de taille *hlen* et qui constitue la première partie du bloque de donnée DB (Data Block). La deuxième portion du DB est le champ PS qui ajoute du bourrage pour avoir toujours un DB de taille fixe. Ce PS est une série de zéros de taille (*nlen* – 2.*hlen* – 2 – *Mlen*) bytes, avec *Mlen* la taille du message X en bytes. La troisième portion est un octet constant '01', et la dernière partie de ce bloque DB, est le message X de taille *Mlen* ≤ *nlen–2.hlen –2*.

La taille de ce DB est ainsi *nlen–hlen–1* bytes et les portions sont concaténés sous cette forme : DB = *HA* ‖ *PS* ‖ 01 ‖ *X*. Après la production de DB, il sera masqué. Pour préparer le masque de ce DB qui s'appelle *dbMask* et qui devra être de même taille que DB (*nlen–hlen–1*), on utilise un MGF qui prend comme entrée un *mgfseed* de taille *hlen* (généré par un RBG= f0) avec un *masklen=nlen–hlen–1* ; *dbMask* = MGF (*mgfSeed*, *nlen–hlen*–1). Notant que le RBG génère une suite d'octets aléatoire de taille *hlen*. Une fois qu'on a le *DB* et le *dbMask*, l'opération XOR sera appliqué sur ces deux bloques (qui sont de même taille) afin d'obtenir le bloque de donnée masqué *MaskedDB* ; *MaskedDB* = *DB* ⊕ *dbMask*. Ce dernier constitue une partie du message EM (Encoded message) qu'il doit être chiffré.

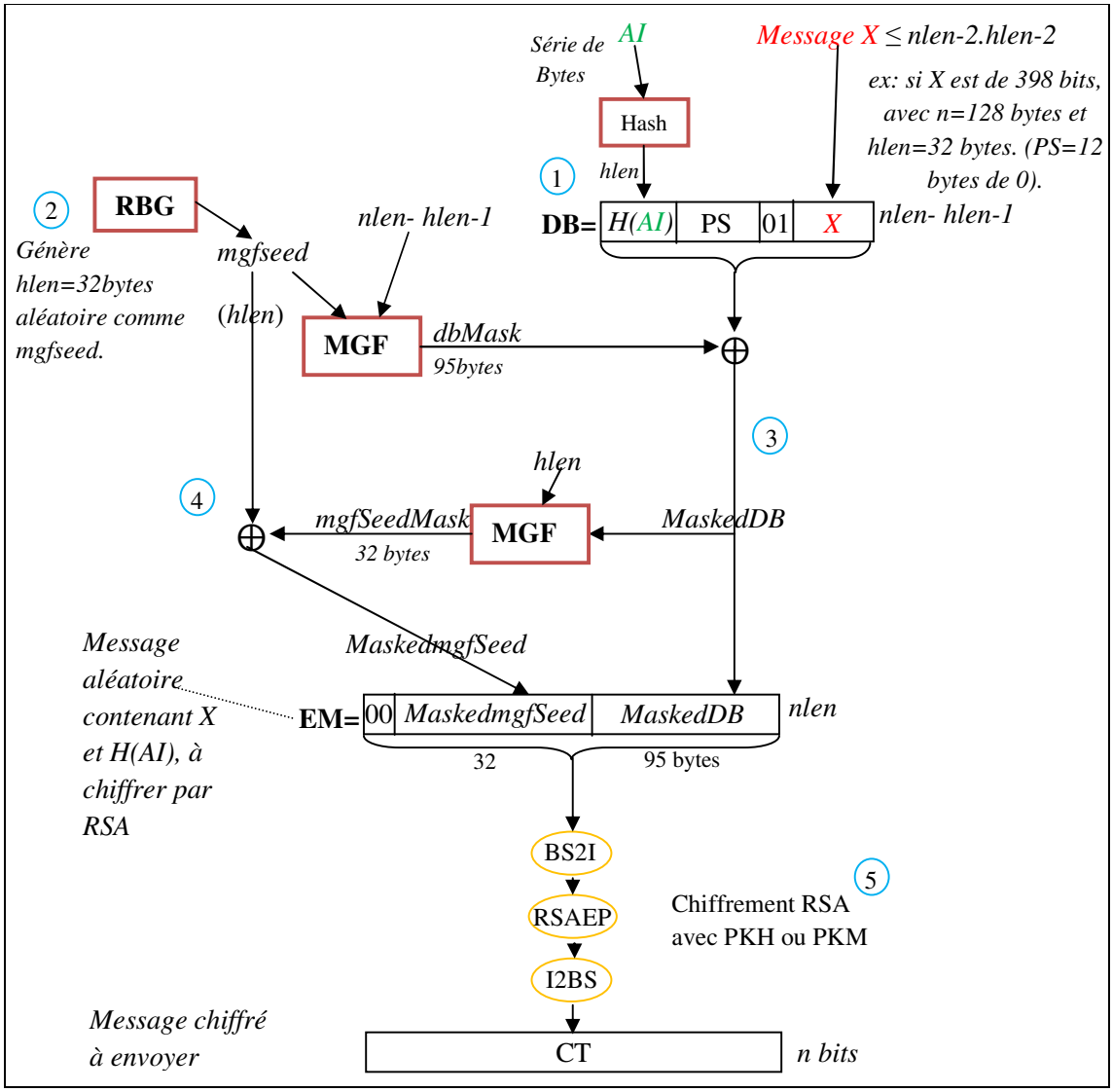

Figure B3.1. Opération de chiffrement RSA-OAEP

Pour que le récepteur puisse extraire le *DB* du *MaskedDB*, il devra savoir le *mgfseed* (généré aléatoirement) pour qu'il puisse générer le même masque *dbMask* et extraire le *DB*. Bien sûr il ne

faut pas transmettre le *mgfseed* en claire. Pour cela le *mgfseed* sera masqué par un *mgfSeedMask* pour accompagner les données dans le message (EM). Le *mgfSeedMask* sera la sortie d'un MGF qui prend comme entrée le *MaskedDB* généré avant, comme entrée aléatoire, et sûrement une taille *masklen* égale à la taille de *mgfseed* qui est *hlen* ; *mgfSeedMask* = MGF (*MaskedDB*, *hlen*). Une fois que le masque de *mgfseed* est disponible, l'opération XOR sera appliqué sur ce *mgfSeedMask* obtenu et le *mgfseed* essentiel afin d'obtenir le *mgfseed* masqué, *MaskedmgfSeed; MaskedmgfSeed=mgfSeed* $\oplus$ *mgfSeedMask*.

Le *MaskedmgfSeed* de taille *hlen* sera la première partie de l'EM, avec la deuxième partie qui est le *MaskedDB* de taille *nlen-hlen-1*, d'où la taille *n-1* bits de l'EM. À la fin un octet fixe «00» sera ajouté à l'entête du EM pour avoir le message EM de taille *n* bits ; *EM* = 00 || *maskedmgfSeed* || *maskedDB*. Ce message EM est la texte en clair qui doit être chiffré par RSAEP (c.à.d. RSA) afin d'obtenir CT= $(EM)^e$ mod *n*.

Notant que le message EM doit être convertir en un entier *em* par la fonction (BS2I, Byte String to Integer) pour qu'il puisse être chiffrer avec RSA en utilisant (*e*, *n*) avec *ct*= $(em)^e$ mod *n*. Ensuite l'entier chiffré *ct* sera convertit par la fonction (I2BS, Integer-to-Byte String) en une suite d'octets chiffrés CT de taille *n*.

Par exemple, si on va utiliser RSA-OAEP pour chiffrer le message A (demande d'attachement) et AI, et on utilise une clé publique RSA de 1024 bits (ou n=128 bytes) avec une fonction de hachage SHA-256 de hlen=256 bits (ou 32 bytes) pour l'utiliser avec MGF et pour l'appliquer sur AI. Dans ce cas la taille maximale permise pour le message A sera (nlen-2hlen-2=128bytes-2.32bytes-2)=62 bytes (ou 496 bits) maximum et qui est suffisant pour les messages A, B et C1 qui ont une taille inférieure ou égale à 50 bytes. Donc si on doit envoyer le message A de 400 bits (ou 50 bytes), il reste 12 bytes (ou 96 bits) pour PS. Dans ce cas la taille du block DB est égale à nlen-hlen-1=128-32-1=95bytes.

B3.2 Déchiffrement RSA-OAEP

A la réception du message chiffré CT, le récepteur (HSS ou MME) utilise son clé privée (SKH ou SKM) pour le déchiffrer. Après l'obtention de EM (voir figure B3.2), il prend la deuxième partie de ce message qui contient le *MaskedDB'* et l'utilise comme entrée à un MGF pour produire le *mgfSeedMask* de taille *hlen*. Une fois qu'il a obtenu ce masque, il applique l'opération XOR de ce *mgfSeedMask* avec la première partie du EM, *Maskedmgfseed'* afin d'obtenir le *mgfseed* qui est l'essentiel pour retrouver le *DB* original. Pour obtenir le masque de *DB* (qui est *dbMask*), Le *mgfseed* déjà obtenu sera l'entrée d'un MGF qui produit ce *dbMask* de taille *nlen-hlen-1*. Une fois qu'on a obtenu ce *dbMask*, on applique l'opération XOR avec le *MaskedDB'* pour obtenir à la fin le *DB*' qui contient le H(*AI*)' et le message X'. Ainsi, le recepteur applique le hachage sur *AI* qui le partage avec l'émetteur, et compare son Hachage H(*AI*) avec H(*AI*)' reçu. Si les deux sont égaux, le récepteur obtient une certaine mesure de l'assurance de l'intégrité des données récupérées.

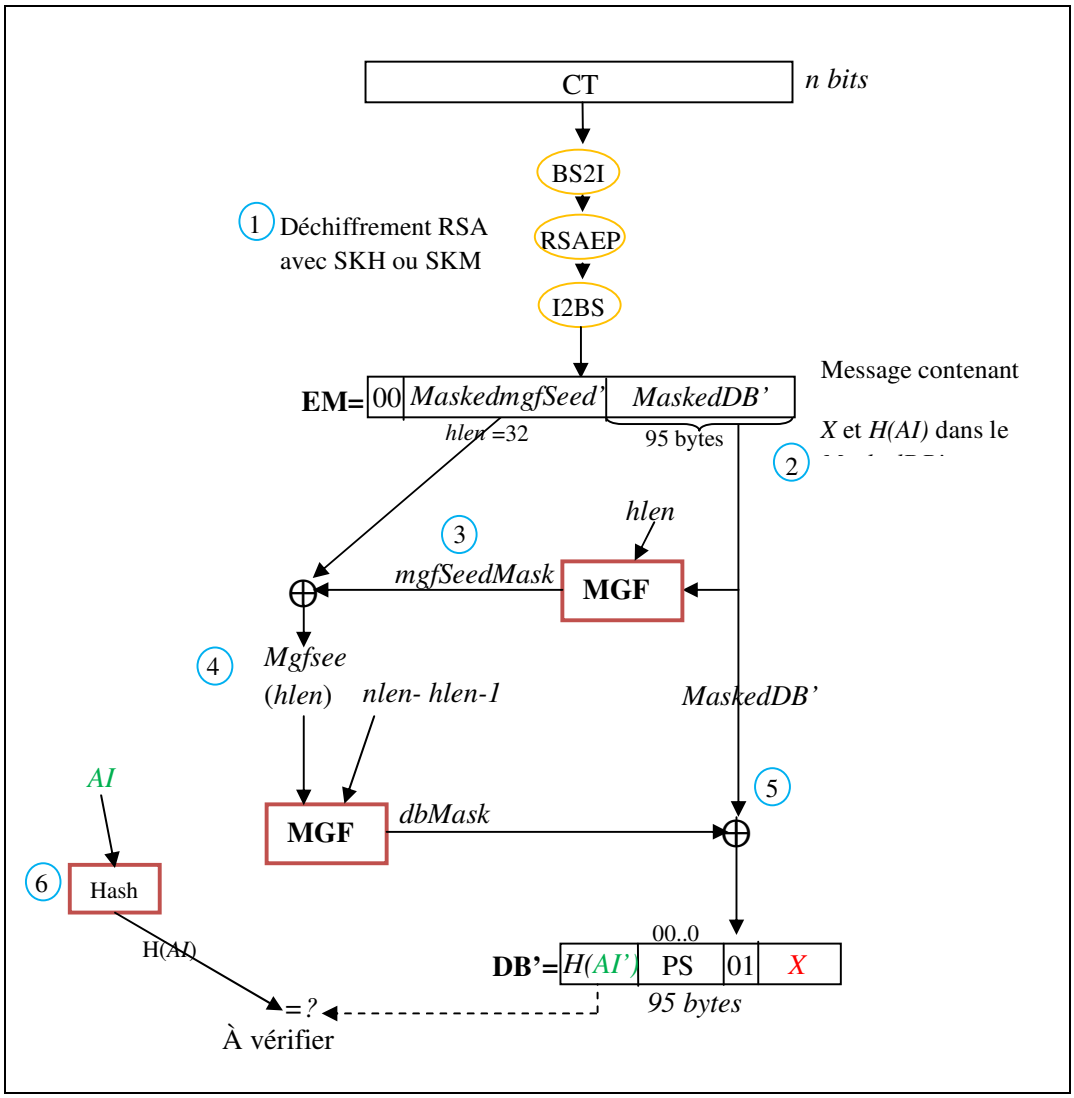

Figure B3.2. Opération de déchiffrement par RSA-OAEP

Références

Références

[**Ahmad *et al.***, 2011] **AHMAD**, **K.,** BAKHACHE, B., EL ASSAD, S., CARAGATA, D. et CHETTO, M. (2011). "Multicast Security Protocol over Satellite DVB Based on Chaotic Sequences", *In Proceedings of the 6th IEEE International Conference for Internet Technology and Secured Transactions, (ICITST-2011)*, Abu Dhabi, UAE, pages 97-102.

[**Ahmad *et al.***, 2012-a] **AHMAD**, **K.,** BAKHACHE, B. et EL ASSAD, S. (2012). "Secure and Efficient Support of Internet Multicast Transmission over Satellite DVB", *In IEEE proceeding of the 3rd International Symposium on Broadband Networks and Fast Internet (RELABIRA'2012)*, Baabda, Lebanon, pages 65-71.

[**Ahmad *et al.***, 2012-b] **AHMAD**, **K.,** BAKHACHE, B., EL ASSAD, S. et SINDIAN, S. (2012). "A Scalable Key Management Scheme for Secure IP Multicast Over DVB-S Using Chaos". *In Proceedings of the 16th IEEE Mediterranean Electrotechnical Conference (MELECON)*, Yasmine Hammamet, Tunisie, pages 736-740.

[**Ahmad *et al.***, 2012-c] **AHMAD**, **K.,** BAKHACHE, B. et EL ASSAD, S. (2012). "A New Security System for IP Multicast Communications over DVB-S", *International Journal of Satellite Communications and Networking*, Accepted Paper.

[Al-Saraireh et *al.*, 2006] AL-SARAIREH, J., YOUSEF, S., et Al NABHAN, M. (2006). "Enhancement Mobile Security and User Confidentiality for UMTS", *In Second European Conference on Mobile Governement*, Brighton, UK, pages 88-93.

[Arapinis *et al*, 2012] ARAPINIS, M., MANCINI, L., RITTER, E., RYAN, M., GOLDE, N., *et al*. (2012). "New Privacy Issues in Mobile Telephony: Fix and Verification". *In Proceeding of ACM conference on Computer and Communications Security CCS*, Raleigh, USA, pages 205-216.

[AVISPA Project, 2013] AVISPA Project, Site: http://www.avispa-project.org.

[Awad *et al.*, 2010] AWAD, A., **AHMAD, K**., EL ASSAD, S. et CARAGATA, D. (2010). "Chaos Based Cryptosystem for Secure Transmitted Images". *In Proceedings of the International Conference on Telecommunications and Multimedia (TEMU)*, Crete, Greece, pages 210-217.

[Bakhache *et al.*, 2011-a] BAKHACHE, B., **AHMAD, K.** et EL ASSAD, S. (2011). "A New Chaotic Encryption Algorithm to Enhance the Security of ZigBee and Wi-Fi networks", *International Journal of Intelligent Computing Research (IJICR)*. Volume 2, numéro 1/2, 2011, pages 219-227.

[Bakhache *et al.*, 2011-b] BAKHACHE, B., **AHMAD, K.** et EL ASSAD, S. (2011). "Chaos based improvement of the security of ZigBee and Wi-Fi networks used for industrial controls". *In Proceedings of the IEEE International Conference on Information Society (i-Society)*, London, UK, pages 139-145.

[Barker et *al.*, 2007] BARKER, E., BARKER, W., BURR, W., POLK, W. et SMID, M. (2007). NIST Special Publication 800-57. Recommendation for Key Management – Part 1: General (Revised).

[Barker *et al.*, 2009] BARKER, E., CHEN, L., REGENSCHEID, A., et SMID, M. (2009). NIST Special Publication 800-56B, Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography.

[Barker et Kelsey, 2012] BARKER, E., et KELSEY, J. (2012). NIST Special Publication (SP) 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators.

[Benchmarks, 2013] Speed Comparison of Popular Crypto++ Algorithms, Available at: http://www.cryptopp.com/benchmarks.html.

[Biham *et al.*, 2005] BIHAM, E., DUNKELMAN, O. et KELLER, N. (2005). "A Related-Key Rectangle attack on the Full KASUMI", *International Association for Cryptologic Research*, Springer-Verlag, ASIACRYPT, pages 443-461.

[Blanchet, 2006] Blanchet, B. (2006). "A Computationally Sound Mechanized Prover for Security Protocols". *In IEEE Symposium o n Security and Privacy*, pages 140–154.

[Bou Abdo *et al.*, 2012-a] BOU ABDO, J., CHAOUCHI, H., et AOUDE, M. (2012). "Ensured Confidentiality Authentication and Key Agreement Protocol for EPS". *In third International Symposium on Broadband Networks and Fast Internet (RELABIRA)*, pages 73-77.

[Bou Abdo *et al.*, 2012-b] BOU ABDO, J., DEMERJIAN, J., et CHAOUCHI, H. (2012). Security v/s Qos for LTE Authentication and Key Agreement Protocol. *International Journal of Network Security & Its Applications (IJNSA)*, Volume 4, numéro 5, pages 71-82.

[Bou Abdo *et al.*, 2013] BOU ABDO, J., DEMERJIAN, J., **AHMAD, K.,** CHAOUCHI, H., et PUJOLLE, G. (2013). "EPS mutual authentication and Crypt-analyzing SPAKA", *In IEEE International Conference on Computing, Management & Telecommunications (ComMan Tel 2013)*, Ho Chi Minh, Vietnam, pages 303-308.

[Bouguen *et al.*, 2012] BOUGUEN, Y, HARDOUIN, E., WOLFF, F. X. (2012). "LTE et les réseaux 4G". Éditions Eyrolles, Paris.

[Caragata *et al.*, 2010] Caragata, D., El Assad, S., Bakhache, B. et Tutanescu, I. (2010). "Secure IP over Satellite DVB Using Chaotic Sequences". *Engineering Letters journal*. Volume 18, numéro 2, pages 135-146.

[Caragata et *al.*, 2011-a] CARAGATA, D., EL ASSAD, S., SHONIREGUN, C., et AKMAYEVA, G. (2011). "UMTS Security: Enhancement of Identification, Authentication and Key Agreement Protocols", In *the 6th International Conference on Internet Technology and Secured Transactions*, pages 278-282.

[Caragata et *al.*, 2011-b] CARAGATA, D., EL ASSAD, S., TUTANESCU, I., SHONIREGUN, C. et AKMAYEVA, G. (2011). "Security of Mobile Internet Access with UMTS/HSDPA/LTE", *In Proceeding of the IEEE World Congress on Internet Security*, London, UK, pages 272-276.

[Caragata, 2011] CARAGATA, D. (2011). "Protocoles de communications sécurisées par des séquences chaotiques. Applications aux standards de communications : IP via DVB-S, UMTS". Thèse de Doctorat de l'université de Nantes et de l'université de Pitesti.

[Chatterjee et Yilmaz, 1992] CHATTERJEE, S. et YILMAZ, M. R. (1992). "Chaos, Fractals and Statistics", *Statistical Science*, Volume 7, numéro 1, pages 49-68.

[Cho *et al.*, 2012] CHO, J. S., KANG, D., KIM, S., OH, J., IM, C. (2012). "Secure UMTS/EPS Authentication and Key Agreement". *Future Information Technology, Application, and Service*, Lecture Notes in Electrical Engineering, Springer Science, Volume 179, pages 75-82.

[Clausen *et al.*, 1999] Clausen H.D., Linder, H. et Collini-Nocker, B. (1999). Internet over direct broadcast satellites. *IEEE Commun. Mag.*, Volume 37, numéro 6, pages 146–151.

[Collini-Nocker et Fairhurst, 2004] COLLINI-NOCKER, B. et FAIRHURST, G. (2004). "ULE versus MPE as an IP over DVB Encapsulation". *In Proceedings of the 2nd International working conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs'04)*, West Yorkshire, U.K.

[Cruickshank *et al.*, 2009] CRUICKSHANK, H., PILLAI, P., NOISTERNIG, M. et IYENGAR, S. (2009). "Security Requirements for the Unidirectional Lightweight Encapsulation (ULE) Protocol". IETF RFC 5458.

[Cruickshank *et al.*, 2008] CRUICKSHANK, H., PILLAI, P., IYENGAR, S. et DUQUEROY, L. (2008). "Security Extension for Unidirectional Lightweight Encapsulation Protocol", IETF Internet Draft.

[Dunkelman *et al.*, 2010] DUNKELMAN, O., KELLER, N. et SHAMIR, A. (2010). "A Practical-time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G telephony", *Proceeding of the 30th annual conference on Advances in Cryptology*. Springer-Verlag Berlin, Volume 6223, pages 393-410.

[Duquerroy *et al.*, 2004] DUQUERROY, L., JOSSET, S., ALPHAND, O., BERTHOU, P. et GAYRAUD, T. (2004). "SatIPSec: an optimized solution for securing multicast and unicast satellite transmissions". *In Proceedings of the 22nd AIAA International Communications Satellite Systems Conference*, Monterey, États Unis, pages 1-11.

[Eastlake, 2008] EASTLAKE, D. (2008). "IANA considerations and IETF protocol usage for IEEE 802 parameters", BCP: 141, RFC 5342.

[EcryptII, 2011-2012] ECRYPT II, European Network of Excellence in Cryptology II, Yearly Report on Algorithms and Keysizes 2011. Disponible sur: < http://www.keylength.com/en/3/>. Dernière visite: Oût 2012.

[El Assad *et al.*, 2008] El Assad, S., Noura, H. et Taralova, I. (2008). "Design and analyses of efficient chaotic generators for crypto-systems". *In Proceedings of the Advances in Electrical and Electronics Engineering- IAENG*, *Special Edition of the World Congress on Engineering and Computer Science (WCECS'08)*, San Francisco, CA, pages 3-12.

[El Assad et Noura, 2011] El ASSAD, S. (85%) et NOURA, H. (15%). (2011). "Generator of chaotic Sequences and corresponding generating system", WO Patent WO/2011/121218.

[El Assad, 2012] El ASSAD, S. (2012). "A new chaos based cryptosystem", internal report, Polytech Nantes, Nantes, 15 pages.

[El Gamal, 1985] El GAMAL, T. (1985). "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *IEEE Transactions on Information Theory*. Volume IT 31, numéro 4, pages 469–472.

[ETSI 301 192, 2004] European Telecommunication Standards Institute (ETSI), EN 301 192 V1.4.1 (2004). *Digital Video Broadcasting (DVB); DVB specification for data broadcasting.*

[ETSI 301 790, 2009] European Telecommunication Standards Institute (ETSI), EN 301 790 V 1.5.1. (2009). *Digital Video Broadcasting (DVB): Interaction Channel for Satellite Distribution Systems.*

[Fairhurst et Collini-Nocker, 2005] FAIRHURST, G. et COLLINI-NOCKER, B. (2005). "Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over MPEG-2 Transport Stream (TS)". IETF RFC 4326.

[Farajallah et *al.*, 2013] FARAJALLAH, M., El ASSAD, S. et CHETTO, M. (2013). "Dynamic Adjustment of the Chaos-based Security in Real-time Energy Harvesting Sensors", *IEEE International conference on Green Computing and Communications*, Beijing, China, 6 pages.

[Farrell, 2000] FARRELL, S. (2000). "The WAP Forum's Wireless Public Key Infrastructure", *Information Security Technical Report*, *Elsevier Science*, Volume 5, numéro 3, pages 23-31.

[Filali *et al.*, 2004] FILALI, F., ANIBA, G. et DABBOUS, W. (2004). "Efficient support of IP Multicast in the Next Generation of GEO satellites". *IEEE Journal on Selected Areas in Communications*, Volume 22, Numéro 2, pages 413-425.

[FIPS 180-4, 2012] Federal Information Processing Standards Publication 180-4. "Secure Hash Standard (SHS)", Information Technology Laboratory, NIST.

[FIPS 186-3, 2009] Federal Information Processing Standards Publication, Specifications for the Digital Signature Standard (DSS), Category: computer security, Subcategory: cryptography, pages 50-61.

[FIPS 197, 2001] National Institute of Standards and Technology, Anouncing the Advanced Encryption Standard. FIPS-197, 2001.

[FIPS 198-1, 2008] Federal Information Processing Standards Publication 198-1. "The Keyed-Hash Message Authentication Code (HMAC)", Information Technology Laboratory, NIST.

[Forsberg *et al.*, 2010] FORSBERG, D., HORN, G., MOELLER, W. D., et NIEMI, V. (2010). "LTE Security". John Wiley & Sons Ltd, United Kingdom.

[Fujisaki *et al.*, 2004] FUJISAKI, E., OKAMOTO, T., POINTCHEVAL, D. et STERN, J. (2004). RSA-OAEP is secure under the RSA assumption, *Journal of Cryptologie*, Volume 17, numéro 2, pages 81-104.

[Glouche *et al.*, 2008] GLOUCHE, Y., GENET, T., et HOUSSAY, E. (2008). SPAN: a Security Protocol ANimator for AVISPA version 1.5, user manual, INRIA/IRISA LANDE Project.

[Harney *et al.*, 2006] HARNEY, H., METH, U., COLEGROVE, A. et GROSS, G. (2006). GSAKMP: Group Secure Association Key Management Protocol. RFC 4535.

[He *et al.*, 2008] HE, D., WANG, J., ZHENG, Y. (2008). "User Authentication Scheme Based on Self-Certified Public-Key for Next Generation Wireless Network". *In IEEE International Symposium on Biometrics and Security Technologies (ISBAST),* pages 1-8.

[Holtmanns *et al.*, 2008], HOLTMANNS, S., NIEMI, V., GINZBOORG, P., LAITINEN, P. et ASOKAN, N. (2008). Cellular Authentication for Mobile and Internet Services, Overview and Application of the Generic Bootstrapping Architecture, John Wiley & Sons, Ltd, Chichester.

[Hong *et al.*, 2005] HONG, T.C., CHEE, W.C. et BUDIARTO R. (2005). "A comparison of IP Datagrams Transmission using MPE and ULE over MPEG/DVB Networks". *In Proceedings of the Fifth International Conference on Information, Communications and Signal Processing*, Bangkok, Thailand, pages 1173-1177.

[Howarth *et al.*, 2004] HOWARTH. M.P., IYENGAR, S., SUN, Z. et CRUICKSHANK, H. (2004). "Dynamics of Key Management in Secure Satellite Multicast". *IEEE Journal on Selected Areas in Communications*. Volume 22, numéro 2, pages 308-319.

[Hubenko *et al.*, 2007] HUBENKO, V.P., RAINES, R.A., BALDWIN R.O, MULLINS, B.E, MILLS, R.F et GRIMAILA, M.R. (2007). "Improving Satellite Multicast Security Scalability by Reducing Re-keying Requirements". *IEEE Network*. Volume 21, numéro 4, pages 51-56.

[IBM, 2013] IBM zEnterprise System, Available at: http://www-03.ibm.com/systems/uk/z/ hardware/ zenterprise/.

[Internet world stats, 2013], the reference site for Internet usage statistics. Disponible sur: <http://www.internetwor ldstats.com/stats.htm>. Dernière visite: Mars 2013.

[Iyengar *et al.*, 2007] IYENGAR, S., CRUICKSHANK, H., PILLAI, P., FAIRHURST, G. et DUQUERROY L. (2007). "Security requirements for IP over satellite DVB networks". *In Proceedings of the 16th IST Mobile and wireless Communications Summit*, Budapest, Hungary, pages 1-6.

[Jonsson et Kalisiki, 2003] JONSSON, J., et KALISKI, B. (2003). RFC 3447, Public-Key Cryptography Standards (PKCS)#1 : RSA cryptography specifications Version 2.1.

[Kaaranen *et al.*, 2005] KAARANEN, H., AHTIAINEN, A., LAITINEN, L., NAGHIAN, S. et NIEMI, V. (2005). "UMTS Networks", Second edition, John Wiley and Sons.

[Khan et *al.*, 2008] KHAN, M., AHMED, A., et CHEEMA, A. R. (2008). "Vulnerabilities of UMTS Access Domain Security Architecture". *In Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, IEEE Computer Society, pages 35-355.

[Kleinjung et *al.,* 2010] KLEINJUNG, T., AOKI, K., FRANKE, J., LENSTRA, A. K., THOMÉ, E., BOS, J. W., GAUDRY, P., KRUPPA, A., MONTGOMERY, P. L., OSVIG, D.A., RIELE, H. te, TIMOFEEV A. et ZIMMERMAN, P. (2010). Factorization of a 768-bit RSA modulus, version 1.4, Cryptology ePrint Archive: Report 2010/006.

[Koblitz, 1987] Koblitz, N. (1987). "Elliptic curve cryptosystems", *Mathematics of Computation*, Volume 48, numéro 177, pages 203-209.

[Kurose et Ross, 2012] KUROSE, J. F. et ROSS, K. W. (2012). "Computer Networking: A Top-Down Approach", 6th edition, Addison-Wesley.

[Lenstra *et al.*, 1993] LENSTRA, A.K., LENSTRA, H.W., Jr., MANASSE, M.S. et POLLARD, J.M. (1993). "The Number Field Sieve", in LENSTRA, A. K. et LENSTRA, H.W., Jr. (eds.) (1993). The

Development of the Number Field Sieve, *Lecture Notes in Mathematics*, Volume 1554, Springer-Verlag, New York, pages 11-42.

[Li *et al.*, 2005] Li, S., Chen, G. et Mou, X. (2005). "On The Dynamical Degradation Of Digital Piecewise Linear Chaotic Maps", *International Journal of Bifurcation and Chaos*, Volume 15, numéro 10, pages 3119-3151.

[Menezes *et al.*, 1997] MENEZES, A. J., OORSCHOT, P.V. et VANSTONE, S. A. (1997). "Handbook of Applied Cryptography", Boca Raton: CRC Press.

[Mjølsnes et Tsay, 2012] MJØLSNES, S. F., et TSAY, J. K.(2012). "Computational Security Analysis of the UMTS and LTE Authentication and Key Agreement Protocols". In Cornell University Library, arXiv:1203.3866.

[Naccache et Stern, 2000] NACCACHE, D. et STERN, J. (2000). Signing on a postcard, *Proceedings of Financial Cryptography FC'00*, LNCS 1962, Springer-Verlag, pages 121-135.

[Ng et Sun, 2005] NG, WHD et SUN, Z. (2005). "Multi-Layers Balanced LKH". *In Proceedings of the IEEE International Conference on Communications (ICC)*, Seoul, Korea, pages 1015-1019.

[NIST, 2013] National Institute of Science and Technology; Information Technology Laboratory; Computer Security Resource Center, http://csrc.nist.gov/groups/ST/hash/sha-3/winner_sha-3.html.

[Pillai et HU, 2006] PILLAI, P. et HU, Y.F. (2006). "Design and Analysis of Secure Transmission of IP over DVB-S/RCS Satellite Systems". *In Proceedings of the third IEEE and IFIP international conference on Wireless and Optical Communications Networks (WOCN)*, Bangalore, India, pages 1 -5.

[Pintsov et Vanstone, 2000] PINTSOV, L.A. et VANSTONE, S.A. (2000). Postal Revenue Collection in the Digital Age: *Proceedings of Financial Cryptography, FC'00*. Lecture Notes in Computer Science 1962, Springer-Verlag, Berlin, pages 105-120.

[Rafaeli et Hutchison, 2003] RAFAELI, S. et HUTCHISON, D. (2003). "A Survey of Key Management for Secure Group Communication". *ACM Computing Surveys*. Volume 35, numéro 3, pages 309-329.

[RFC 2401, 1998] KENT, S. et ATKINSON, R. (1998). RFC2401, "Security Architecture for the Internet Protocol".

[RFC 4301, 2005] KENT, S. et SEO, K. (2005). "Security Architecture for the Internet Protocol".

[RFC 4303, 2005] Kent, S. (2005). RFC4303, "IP Encapsulating Security Payload (ESP)".

[RFC 4306, 2005] Internet Engineering Task Force (IETF), 2005. *RFC 4306, Internet Key Exchange (IKEv2) Protocol.* Kaufman C.

[Rivest *et al.,* 1978] RIVEST, R.L., SHAMIR, A., et ADLEMAN, L.M. (1978). "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, Volume 2, numéro 21, pages 120-126.

[Rohrer *et al.*, 2007] ROHRER, J. P., STERBENZ, J. P.G., et WEICHAO, W. (2007). "Homogeneous Security in Heterogeneous Networks: Towards A Generic Security Management Protocol". *IEEE Military Communications Conference (MILCOM)*, pages1-6.

[Strobel, 2007] STROBEL., D. (2007). "IMSI Catcher". Seminar Work, Ruhr-Universitat Bochum.

[Tien Thinh, 2010] TIEN THINH, N. (2010). "Evolution de la couche RRC de la plateforme OpenAir vers les nouvelles normes LTE", *Mémoire de fin d'études de Master en informatique*, Institut de la francophonie pour l'informatique, Sophia-Antipolis, 48 pages.

[TR, 2011] Technical Report, Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 4: Design and Evaluation Report., version 2, 2011.

[TR 33.821, 2009] 3GPP Technical Report 33.821, V.9.0.0, Technical Specification Group Services and System Aspects; Rationale and track of security decisions in Long Term Evolved (LTE) RAN / 3GPP System Architecture Evolution (SAE).

[TS 22.048, 2003] 3GPP TS 22.048, Digital cellular telecommunications system (Phase 2+), UMTS, "Security Mechanisms for the (U)SIM application toolkit"; Stage 1

[TS 22.101, 2009] 3GPP TS 22.101, UMTS, "Service aspects, Service principles".

[TS 22.278, 2008 ou 2012] 3GPP TS 22.278, UMTS, LTE, "Service Requirements for the Evolved Packet System (EPS)".

[TS 23.048, 2005] 3GPP Technical Specification Group Services Core Network and Terminals, "Security mechanisms for the (U)SIM application toolkit".

[TS 23.060, 2011] 3GPP TS 23.060, Digital cellular telecommunications system (Phase 2+), UMTS, General Packet Radio Service (GPRS), "Service description", Stage 2.

[TS 23.401, 2012] 3GPP TS 23.401, "LTE, General Packet Radio Service (GPRS) Enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Access".

[TS 24.301, 2008-2011] 3GPP TS 24.301, UMTS, LTE, "Non-Access-Stratum (NAS) Protocol for Evolved Packet System (EPS)", Stage 3

[TS 31.101, 2011] 3GPP TS 31.101, UMTS, "UICC-terminal interface, Physical and logical characteristics".

[TS 31.111, 2009] 3GPP TS 31.111, Digital cellular telecommunications system (Phase 2+), UMTS, LTE, "Universal Subscriber Identity Module (USIM) Application Toolkit (USAT)".

[TS 33.102, 2012] 3GPP TS 33.102, Technical Specification Group Services and System Aspects; "3G Security; Security architecture".

[TS 33.103, 2001] 3GPP TS 33.103, Technical Specification Group Services and System Aspects, "3G security, Integration guidelines".

[TS 33.120, 1999] 3GPP TS 33.120, "3G Security, Security Principles and Objectives".

[TS 33.210, 2010] 3GPP TS 33.210, Technical Specification Group Services and System Aspects; "3G Security; Network Domain Security (NDS); IP Network Layer Security".

[TS 33.220, 2012] 3GPP TS 33.220, “Generic Authentication Architecture (GAA), Generic Bootstrapping Architecture (GBA)”.

[TS 33.310, 2010] 3GPP TS 33.310, Technical Specification Group Services and System Aspects; “Network Domain Security (NDS), Authentication Framework (AF)”.

[TS 33.401, 2012] Digital cellular telecommunications system (Phase 2+), UMTS, LTE, 3GPP System Architecture Evolution (SAE), Security architecture, Release 11.

[TS 35.201, 2009] 3GPP TS 35.201, Technical Specification Group Services and System Aspects; “3G security, Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 1: f8 and f9 specifications”.

[TS 35.202, 2009] 3GPP TS 35.202, Technical Specification Group Services and System Aspects; “3G security; Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2: Kasumi specification”.

[TS 35.215, 2012] 3GPP TS 35.215, Technical Specification Group Services and System Aspects, “Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 and UIA2, Document 1: UEA2 and UIA2 Specifications”.

[TS 35.216, 2012] 3GPP TS 35.216, Technical Specification Group Services and System Aspects; “Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 2: SNOW 3G Specification”.

[TS 35.221, 2012] 3GPP TS 35.221, “Confidentiality and Integrity Algorithms EEA3 and EIA3”, Document 1: EEA3 and EIA3 Specifications.

[TS 36.300, 2008-2012] 3GPP TS 36.300, “Evolved Universal Terrestrial Radio Access and Evolved Universal Terrestrial Radio Access Network (E-UTRAN)”, Overall Description, Stage 2.

[TS 36.323, 2012] 3GPP TS 36.323, “Evolved Universal Terrestrial Radio Access (E-UTRA), Packet Data Convergence Protocol (PDCP) Specification”.

[TS 36.331, 2011] 3GPP TS 36.331, “E-UTRA, Radio Resource Control, Protocol Specification”.

[TS 43.020, 2011] 3GPP TS 43.020, Digital cellular telecommunications system (Phase 2+), “Security-Related network functions”.

[ITU, 2012] ITU Telecommunications, Global mobile-cellular subscriptions, total and per 100 inhabitants, 2001-2011, dernier accès: décembre 2012.

[Walker, 2013] WALKER, C. (2013). Feds’ New Cell Phone Spying Device Raising Privacy Concerns, Available at: http://www.thedailychronic.net/2013/16934/feds-new-cell-phone-spying-device-raising-privacy-concerns/.

[Xiehua *et al.*, 2011] XIEHUA, L., et YONGJUN, W. (2011). “Security Enhanced Authentication and Key Agreement Protocol for LTE/SAE Network”. *In 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pages 1-4.

[Yavuz *et al.*, 2006] YAVUZ, A., ALAGOZ, F. et ANARIM, E. (2006). “A new Satellite Multicast Security Protocol Based on Elliptic Curve signatures”. *In Proceedings of the IEEE International Conference on Information and Communication Technologies (ICTTA’06)*, pages 2512-2517.

Liste des Sigles

Liste des Sigles

3G	3ème Génération de réseaux mobiles
3GPP	3rd Generation Partnership Project
4G	4ème Génération de réseaux mobiles
AA	Authentication Algorithm
AAL5	ATM Adaptation Layer 5
AES	Advanced Encryption Standard
AES/CMAC	AES Cipher-based Message Authentication Code
AES/CTR	AES Counter Mode
AH	Authentication Header
AI	Additional Information
AK	Anonymity Key
AKA	Authentication and Key Agreement
AMF	Authentication and key Management Field
AS	Access Stratum
ASME	Access Security Management Entity
ATM	Asynchronous Transfer Mode
AuC	Authentication Center
AUTN	Authentication Token
AV	Authentication Vector
AVISPA	Automated Validation of Internet Security Protocols and Applications
BS2I	Byte String to Integer
BSC	Base Station Controller
BTS	Base Transceiver Station
CA	Certification Authority
CAPEX	Capital Expenditure
CBC	Cipher-Block Chaining
CDMA	Code Division Multiple Access
CERT	Certificate
CK	Ciphering Key
CMS	Commande du mode de sécurité

CoreBW	Core Bandwidth
CR	Certificate Request
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
C-RNTI	Cell Radio Network Temporary Identity
CS	Circuit Switched
CT	Cipher Text
DB	Data Block
DDoS	Distributed DoS
DeNB	Donor eNB
DO	Data Overhead
DoS	Denial of Service
DSA	Digital Signature Algorithm
DULM	Data Unit Labelling Method
DVB	Digital Video Broadcasting
DVB-C	DVB Cable
DVB-RCS	DVB Return Channel via Satellite
DVB-S	DVB Satellite
DVB-SH	DVB Satellite Handheld
DVB-T	DVB Terrestrial
EARFCN-DL	E-UTRA Absolute Radio Frequency Channel Number-Down Link
EBD	Effective Bandwidth Downlink
EBU	Effective Bandwidth Upload
EC-AKA	Ensured Confidentiality-AKA
ECC	Elleptic Curve Cryptography
ECDSA	Elleptic Curve DSA
ECPVSS	Elleptic Curve Pintsov-Vanstone Signature Scheme
EDGE	Enhanced Data rates for GSM Evolution
EEA	EPS Encryption Algorithm
EEMSUCU	Enhanced EMSUCU
EH	Extension Header

EIA	EPS Integrity Algorithm
EIR	Equipement Identity Register
EK	Encryption Key
eKSI	evolved KSI en EPS
EM	Encoded Message
EMM	EPS Mobility Management
EMSUCU	Enhancement Mobile Security and User Confidentiality
eNB	evolved NodeB
EPC	Evolved Packet Core
EPS	Evolved Packet System
ESP	Encapsulating Security Payload
ETSI	European Telecommunications Standards Institute
EULE	Enhanced ULE
E-UTRAN	Evolved UTRAN
FC	Function Code
FEC	Forward Error Correction
FIPS	Federal Information Processing Standard
FKH	Fixed KPDU Header length
FP-AKA	Full Protection-AKA
GEO	Geostationary Earth Orbit
GERAN	GSM/EDGE Radio Access Network
GGSN	Gateway GPRS Support Node
GK	Group Key
GKC	Group Key Controller
GM	Group Member
GMSC	Gateway MSC
GPRS	General Packet Radio Service
GSE	Generic Stream Encapsulation
GSM	Global System for Mobile Communications
GUMMEI	Globally Unique MME Identifier
GUTI	Globally Unique Temporary UE Identity

GW	Gateway
HCS	Header Check Sum
HE	Home Environment
HFN	Hyper Frame Number
HLPSL	High-Level Protocol Specification Language
HLR	Home Location Register
HMAC	Keyed-Hash Message Authentication Code
HSS	Home Subscriber Server
I2BS	Integer-to-Byte String
IANA	Internet Assigned Numbers Authority
ID	Identity
IE	Information Element
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IK	Integrity Key
IKE	Internet Key Exchange
IMEI	International Mobile Equipment Identity
IMEISV	IMEI and Software Version Number
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPPRU	IP Packet Recovery Unit
IPsec	IP Security
IS-95	Interim Standard 95
ISAKMP	Internet Security Association Key Management Protocol
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
ITU	International Telecommunication Union
K_{ASME}	Local Master Key
KDF	Key Derivation Function
KE	Key Exchange
KEK	Key Encryption Keys

K_{eNB}	Intermediate Key at eNB Level
KIS	Key ID Size
KL	Key Length
KMD	Key Management Data
KPDU	Key PDU
KRH	Private Key of HSS
KRM	Private Key of MME
KSI	Key Set Identifier en 3G
KV	Key Version
KVS	Key Version Size
LAI	Location Area Identity
LAN	Local Area Network
LFSR	Linear Feedback Shift Register
LKH	Logical Key Hierarchy
LTE	Long Term Evolution
MAC	Medium Access Control
MAC	Message Authentication Code
MAC-I	Message Authentication Code for Integrity
MAPSec	Mobile Application Part Security
MCC	Mobile Country Code
ME	Mobile Equipment
MGF	Mask Generation Function
MHEK	MME-HSS shared Encryption Key
MHIK	MME-HSS shared Integrity Key
MHK	MME-HSS shared Key
MIPS	Million Instructions Per Second
MITM	Man in the Middle
MME	Mobility Management Entity
MNC	Mobile Network Code
MPE	Multi Protocol Encapsulation
MPEG	Moving Pictures Expert Group

MPEG-2 TS	MPEG-2 Transport Stream
MS	Mobile Station
MSIN	Mobile Subscriber Identification Number
MSC	Mobile Switching Center
NAS	Non-Access Stratum
NCC	Network Control Center
NIST	National Institute of Standards and Technology
NPA	Network Point of Attachment
NSQN	New SQN
NT	Number of Transmitted packets
OBP	On-Board Processor
OBS	On Board Switching
OMAC	One-key CBC MAC
OPEX	Operational Expenditure
PCI	Physical Cell Identity
PCRF	Policy Control and Charging Rules Functions
PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
PDN GW	PDN Gateway
PDU	Protocol Data Unit
PEK	Permanent Encryption Key
PID	Packet Identifier
PIK	Permanent Integrity Key
PIM	Protocol Independent Multicast
PIN	Personal Identification Number
PKH	Public Key of HSS
PKI	Public Key Infrastructure
PKM	Public Key of MME
PKU	Public Key of UE
PN	Packet Number
PRNG	Pseudo Random Number Generator

PS	Packet Switched
PSQN	Previous SQN
P-TMSI	Packet TMSI
PUSI	Payload Unit Start Indicator
PWLCM	Piecewise Linear Chaotic Map
QoS	Quality of Service
RAI	Routing Area Identity
RAND	Random value
RandEK	Random Encryption Key
RandHM2	Random value generated by HSS
RandIK	Random Integrity Key
RandMH1	Random value generated by MME
RBG	Random Bit Generator
RCST	Return Channel via Satellite Terminal
RES	RESponse
RFC	Request For Comments
RLC	Radio Link Control
RN	Relay Node
RNC	Radio Network Controller
RRC	Radio Resource Control
RRM	Radio Resource Management
RSA	Rivest Shamir Adleman
RSAEP	RSA Encryption Primitive
RSADP	RSA Decryption Primitive
RSA-OAEP	RSA- Optimal Asymmetric Encryption Padding
SA	Security Association
SAD	Security Association Database
SAE	System Architecture Evolution
SDO	Standards Development Organizations
SE-AKA	Security Enhanced-AKA
SEULE	Secured EULE

SGSN	Serving GPRS Support Node
S-GW	Serving Gateway
SHA	Secure Hash Algorithm
SLS	Switching-Label Size
SMAP	Satellite Multicast Adaptation Protocol
SN	Sequence Number
SNDU	Sub-Network Data Unit
SN id	Serving Network identity
SPD	Security Policy Database
SPI	Security Parameter Index
SQN	Sequence Number
SRB	Signalling Radio Bearer
S-TMSI	S-Temporary Mobile Subscriber Identity
TAI	Tracking Area Identifier
TAU	Tracking Area Update
TEK	Transient Encryption Key
TIK	Temporary Integrity Key
TIK	Transient Integrity Key
TK	Transient Key
TLKH	Two-Tiered LKH
TLS	Transport Layer Security
TMSI	Temporary Mobile Subscriber Identity
TNK	Total Number of Keys
TR	Technical Report
TS	Technical Specification
TTP	Trusted Third Party
UE	User Equipment
UEA	UMTS Encryption Algorithm
UESecCapab	UE Security Capabilities
UIA	UMTS Integrity Algorithm
UICC	Universal Integrated Circuit Card

UK	Unique Key
ULE	Unidirectional Lightweight Encapsulation
ULE Sec-ID	ULE Security Identifier
UMEK	UE-MME shared Encryption Key
UMIK	UE-MME shared Integrity Key
UMTS	Universal Mobile Telecommunications System
UP	User Plane
USIM	Universal Subscriber Identity Module
UTRAN	Universal Terrestrial Radio Access Network
VID	Vendor ID
VLR	Visitor Location Register
VOIP	Voice Over IP
VPN	Virtual Private Networks
XMAC-I	Expected MAC-I
XRES	Expected Response
ZuC	Zu Chongzhi

Université Libanaise
École Doctorale
Sciences et Technologies

l'université nantes angers le mans
PÔLE DE RECHERCHE ET D'ENSEIGNEMENT SUPÉRIEUR

UNIVERSITÉ DE NANTES

Thèse de Doctorat

Kassem AHMAD

Protocoles, gestion et transmission sécurisée par chaos des clés secrètes. Applications aux standards: TCP/IP via DVB-S, UMTS, EPS.

Protocols, management, and secured transmission by chaos of the secret keys. Applications in the standards: TCP / IP via DVB-S, UMTS, EPS

Résumé

IP Multicast est supporté dans la nouvelle génération des systèmes satellitaires implémentant DVB-S (Digital Video Broadcasting via satellite). Dans ce type de communication, la sécurité, la commutation, et l'évolutivité sont les principaux défis. A ce propos, nous proposons un nouveau système de sécurité multicast basé sur : une méthode d'encapsulation améliorée du standard ULE qui peut opérer avec les approches de commutation 'label ou self-switching' afin d'assurer : un transfert efficace d'IP multicast, un mécanisme de sécurité très performant, et un système évolutif de gestion de clés à deux couches LKH (Logical Key Hierarchy). L'utilisation du chaos est proposée pour la génération de nouvelles clés et le chiffrement des données. L'analyse du système proposé montre qu'il peut gérer un très grand nombre des membres d'une manière sécurisée et efficace avec une consommation minimale de bande passante.

La sécurité dans les réseaux mobiles de 4ème génération EPS est considérée comme très robuste. Mais, des failles héritées de l'UMTS et d'autres identifiées dans la littérature spécialisée restent sans traitement efficace. Ces vulnérabilités affectent précisément le protocole d'authentification et d'établissement des clés, l'EPS-AKA. Plusieurs protocoles ont été proposés pour résoudre ces problèmes mais sans réussite significative. Dans cette optique, nous proposons un nouveau protocole appelé FP-AKA qui assure une forte protection contre les différentes attaques avec un coût minimal. La comparaison de FP-AKA avec les meilleurs protocoles existants dans la littérature (SE-AKA, EC-AKA,..) montre la supériorité de FP-AKA au niveau de plusieurs paramètres (sécurité, coût, délai,..).

Mots clés

Sécurité, DVB-S, Unidirectional Lightweight Encapsulation (ULE), Multicast, LKH, Gestion des clés, Séquences Chaotiques, Universal Mobile Telecommunications System (UMTS), Evolved Packet System (EPS), AKA, LTE, Attaques.

Abstract

IP multicast is supported in the next generation of satellite systems implementing DVB-S (Digital Video Broadcasting via Satellite). In this type of communication, security, switching and scalability are the main challenges. In this context, we propose a new multicast security system based on: an enhanced ULE encapsulation standard, method which can operate with the switching approaches 'label or self-switching' to ensure efficient filtering and multicast forwarding, a highly flexible security mechanism, and a scalable key management scheme with two LKH (Logical Key Hierarchy) layers. The usage of chaos is proposed for the new keys generation and data encryption. The analysis of the proposed system shows that it can handle a large number of members in a secure and efficient manner with minimal bandwidth consumption.

Security in the 4th generation of mobile networks EPS is considered very robust. However, weaknesses inherited from UMTS and others identified in the specialized literature remain without effective treatment. These vulnerabilities affect precisely the authentication and key agreement protocol, EPS-AKA. Several protocols have been proposed to resolve these problems but without a significant success. In this context, we propose a new protocol called FP-AKA which provides a strong protection against the different attacks with minimal cost. The comparison of FP-AKA with the best existing protocols in the literature (SE-AKA, EC-AKA,..) shows the superiority of FP-AKA in several parameters (security, cost, delay,..).

Key Words

Security, DVB-S, Unidirectional Lightweight Encapsulation (ULE), Multicast, LKH, Key Management, Chaotic Sequences, Universal Mobile Telecommunications System (UMTS), Evolved Packet System (EPS), AKA, LTE, Attacks.